Vignette for `CusumActMgr` Function

Chindhanai Uthaisaad

August 28, 2017

Abstract

This vignette describes the use of new function `cusumActMgr` in the R package `factorAnalyticsAddons`, implemented during the Google Summer of Code (GSoC) 2017 *Advancing factorAnalytics* project. GSoC funding for this project was awarded to UW AMATH MS-CFRM student Chindhanai Uthaisaad, with Douglas Martin and Thomas Philips as mentors.

1 Using Statistical Process Control to Monitor Active Managers

The methodology implemented in the new function `cusumActMgr` was introduced in Philips, Yashchin and Stein (2003), and further details were supplied in the unpublished more technical version of the paper. For the reader's convenience, the key parts of those authors' work are summarized in this section.

Most traditional measures of portfolio performance (e.g. Excess Return, Sharpe Ratio and Information Ratio) suffer from a serious limitation on account of the fact that they are computed over a fixed, pre-determined, interval: good performance in some years can conceal poor performance in others, making it difficult to assess a portfolio's current performance and harder still to identify transitions from good performance to bad. Indeed, it is often claimed that it takes 40 years, based on the results of a t-test (t>2), to determine whether a portfolio has outperformed its benchmark. While this is true if the portfolio's active returns (returns in excess of its benchmark) are stationary, it is of little use to investors, who should not (and do not) wait 40 years to determine if their actively managed portfolio is outperforming its benchmark.

In fact, time variation in the mean excess return is the norm in an efficient market. As inefficiencies appear and are then competed away, all investment processes will at some times flourish and at other times underperform, necessitating a fundamentally different approach to performance measurement: investors ought to continually re-estimate the *current* performance of their portfolios, and should rigorously re-evaluate a manager's strategy as soon as they can determine with some confidence that it is no longer adding value. Classical statistics does not offer good ways in which to make such a determination. However, there is a rich body of work on change point detection that is closely related to sequential analysis and statistical process control (SPC), and which can be applied profitably to this problem.

The Cusum (an acronym for "cumulative sum") is a change point detection procedure that is particularly well suited to the problem of rapidly detecting changes in the mean of a noisy random process, as it is robust to the distribution of portfolio returns, and, remarkably, offers an optimal trade-off between detection speed and the rate of false alarms. For a wide range of distributions in the exponential family (i.e. distributions of the form $h(x) \cdot g(\theta) \cdot \sum_i w_i(\theta) \cdot t_i(x)$, which includes the normal, log-normal and gamma), and for any given rate of false alarms, the Cusum detects an abrupt change in the mean faster than any other method. When applied to actively managed portfolios, it allows us to detect under-performance with reasonable confidence in about 40 months. The Cusum implements a Likelihood Ratio Test that is closely related to Wald's Sequential Probability Ratio test (SPRT), and requires three inputs:

1. A measure of performance that is to be monitored for an abrupt change in its mean.
2. The value of the measure when the portfolio's performance is acceptable
3. The value of the measure when the portfolio's performance is unacceptable.

We use a special estimator of the information ratio IR as our measure of performance, where the true but unknown value of the information ratio is defined as

$$IR = \frac{\mu_A}{\sigma_A} = \frac{E(r_A)}{var^{1/2}(r_A)}$$

where $r_A = r_P - r_B$ is the *active return* of the portfolio P with return r_P relative to the benchmark B with return r_B. The standard deviation σ_A is called the *tracking error.*

The standard estimate of the information ratio based on a historical set $r_{A,1}, r_{A,2}, \cdots, r_{A,N}$ of active returns is

$$\widehat{IR}_N = \frac{\hat{\mu}_{A,N}}{\hat{\sigma}_{A,N}}$$

where $\hat{\mu}_{A,N}$ and $\hat{\sigma}_{A,N}$ are the sample mean and sample standard deviation of the active returns.[1] However, the method we use is based on a special semi-instantaneous version of information ratio estimator called the "current" information ratio estimator, that is described below.

We define an annualized information ratio of 0.5 to be an acceptable level of performance, as, in our experience, active managers seldom produce an *IR* higher than 0.5 for an extended period of time. Likewise, we define an annualized *IR* of 0 to correspond to bad performance, as an index fund can provide similar performance at lower cost.

In outline the Cusum procedure works as follows: Every month, after new returns are recorded for the portfolio and its benchmark, its active return is computed, and the most recent active return, along with all the past returns, are used to construct a particular, backward looking, form of Wald's Sequential Probability Ratio Test (SPRT) that discriminates between two hypotheses:

1. H_0: The portfolio's current annualized information is 0.5
2. H_1: The portfolio current annualized information is 0

When sufficient evidence accrues to reject H_0 (i.e. the likelihood ratio becomes sufficiently high), an alarm is raised, triggering a thorough investigation into the causes of underperformance.

We estimate the current information ratio as follows:

1. Compute the current log active return of the portfolio relative to its benchmark

$$e_i = \log\left(\frac{V_{P,i}}{V_{B,i}}\right) = \log\left(\frac{1+r_i}{1+b_i}\right) \tag{1}$$

[1] Both IR and $\hat{IR}$ have the convenient feature of being scale invariant, i.e., there values do not change if r_A or the values of an observed data set $r_{A,1}, r_{A,2}, \cdots, r_{A,N}$ are multiplied by an arbitrary positive constant. Furthermore, in the case of uncorrelated normally distribution active returns with mean μ and standard deviation σ the quantity $\sqrt{N} \cdot \widehat{IR}_N$ has a t-distribution with $N-1$ degrees of freedom, and is well approximated by a standard normal distribution in most applications.

where $V_{P,i}$ is the value of the portfolio P in the i^{th} month, $V_{B,i}$ is the values of the benchmark B in the i^{th}month, r_i is the arithmetic return of the portfolio in the i^{th}month, and b_i is the arithmetic return of the benchmark in the same month. Taking logs allows us to control naturally for the impact of volatility on compounding, as geometric returns are, ultimately, what matter to investors.

2. Estimate the current tracking error of the portfolio using a robust exponentially weighted estimator of volatility. For many problems in finance, the mean return is close to zero, so that the variance and the second moment are about the same. A simple exponentially weighted estimate of volatility can therefore be written as follows:

$$\hat{\sigma}_i^2 = \gamma \hat{\sigma}_{i-1}^2 + 12(1-\gamma)e_i^2 \quad \text{for } i = 2, 3, \ldots, \ n. \tag{2}$$

Unfortunately, this estimator is not robust to outliers. If we experience an exceptionally large excess return, the estimate of variance will rise sharply, and the recursion will overestimate risk for a long time. Also, if there is a sharp shift in the level of risk, either up or down, the exponentially weighted estimator will take a long time to adapt to the new regime. We address this issue in two ways:

(a) Huberize the returns at $k\sigma$ (we set $k = 4$), i.e., set the return at $\pm k\sigma$ depending on the sign of the return, whenever the absolute value of the return is greater than $k\sigma$.[2] This avoids over-estimation of the level of risk in the event of an outlier. By Chebychev's inequality, the probability of exceeding this level is $< 1/4^2 = 0.0625$

(b) Lowerγ by about 10% if $|r_i| > 4\sigma_{i-i}$ or $|r_i| < \sigma_{i-1}/4$ so that we adapt more quickly to changes in the level of risk.

These insights can be combined as follows:

$$\hat{\sigma}_i^2 = \gamma_i \hat{\sigma}_{i-1}^2 + 12(1-\gamma_i)(e_i')^2 \quad \text{for } i = 2, 3, \ldots, \ n. \tag{3}$$

where $0 < \gamma < 1$, and

$$e_i' = sgn(e_i) \cdot \min(|e_i|, 4\sigma). \tag{4}$$

We set$\gamma_i = 0.8$ if $e_i < 0.25\sigma_{i-1}$ or $e_i > 4\sigma_{i-1}$, and set$\gamma_i = 0.9$ otherwise. These values are chosen based on experience, and provide a good balance between detection speed and protection against undue outlier influence.

[2]This action is called "Huberization" after the formula of Peter Huber's robust M-estimator of location. See Equation (2.28) and the lower plot in Figure 2.3 of the Maronna, Martin and Yohai (2006) book *Robust Statistics: Theory and Methods.*

3. Compute the portfolio's current IR estimator using the newly recorded return:

$$\widehat{IR}_i = \frac{\sqrt{12}\, e_i}{\hat{\sigma}_{i-1}}, \tag{5}$$

Note that this is a semi-instantaneous estimator of IR in that it uses the current return e_i in the numerator and the historical volatility estimator $\hat{\sigma}_{i-1}$ based on past returns in the denominator. The use of $\hat{\sigma}_{i-1}$, instead of $\hat{\sigma}_i$ in the denominator $\widehat{IR}_i$ ensures that the current estimate $\widehat{IR}_i$ of IR has the following properties: (a) It is approximately unbiased and uncorrelated, and (b) it has approximately a standard normal distribution when the e_i have a normal distribution. The latter property allows for using a z-test that the current IR estimate $\widehat{IR}_i$ is an outlier.

We now use the sequence of current estimates $\widehat{IR}_i$ to detect when the true IR has changed. Each time a new return is recorded, the Cusum updates a backward looking SPRT against these two alternatives ($H_0 : IR = 0.5$ and $H_1 : IR = 0$). In effect, it determines an optimal estimation interval over which to compute the likelihood ratio for these two alternatives, and then performs the computation. In practice, it proves easier to compute the logarithm of the likelihood ratio, and it can be shown that the log-likelihood ratio satisfies the following recursion:

$$\begin{aligned} L_0 &= 0 \\ L_N &= \max\left\{0,\ L_{N-1} - \widehat{IR}_N + 0.25\right\} \quad \text{for } N = 1, 2, \ldots \end{aligned} \tag{6}$$

We monitor L_N, and when it exceeds a user-defined threshold, we raise an alarm, as sufficient statistical evidence has accrued to suggest that the manager's performance has deteriorated. In practice, a threshold setting that detects flat-to-benchmark performance in three and a half years and allows one false alarm in seven years proves satisfactory. This setting was determined by examining the real investment results of a large number of domestic and international equity and fixed income managers employed by a major corporate pension fund. It affords a tenfold improvement in detection speed over the traditional t-test for a difference in means while still maintaining a reasonably low rate of false alarms. Table 1 shows the threshold required for various expected times between false alarms.

Even though we do not use $IR = -0.5$ in our hypothesis test, we include the run lengths associated with it in Table 1 to give the user a broader perspective on the functional relationship between the average run length, the Threshold and the Information Ratio. It is important to note that the Cusum procedure cannot provide a causal explanation for underperformance.

Thresholds for L_N	Expected time to cross the Threshold (months)		
	IR = 0.5;T_0	IR = 0;T_1	IR=-0.5;T_2
11.81	24	16	11
15.00	36	22	15
17.60	48	27	18
19.81	60	32	21
21.79	72	37	23
23.59	84	41	25

Table 1: The threshold required for various expected times between false alarms.

It falls to the investor to make an independent determination of whether or not the alarm was raised on account of a weakness in the investment process or is a false positive raised on account of the natural ebb and flow of performance that afflicts all investment processes from time to time. .

2 The Cusum Algorithm and R Code for Monitoring Active Managers

2.1 The Computational Algorithm

The Cusum algorithm is implemented using the following steps:

1. Initialize the tracking error σ_0 to an estimate of its expected value, typically the sample volatility over an appropriate historical period, and set $L_0 = 0$.
2. Each time a new return is recorded:
   (a) Compute the logarithmic active return using Equation (1)
   (b) Update the estimate of tracking error using Equations (3) and (4)
   (c) Estimate the current information ratio using Equation (5)
   (d) Update the likelihood ratio using Equation (6)
3. Compare the likelihood ratio to the thresholds in Table (1)
   (a) If it does not exceed the user-chosen threshold, do nothing, and go back to step 2.

(b) If it exceeds the threshold, raise an alarm and investigate the manager's process to determine the cause of underperformance

   i. If the investment process is found satisfactory, consider this to have been a false alarm. Reset L_N to 0, and continue monitoring the portfolio starting with step 2.

   ii. If the investment process is found deficient, stop the Cusum and decide on a course of corrective action

2.2 The R Function cusumActMgr and Its Use

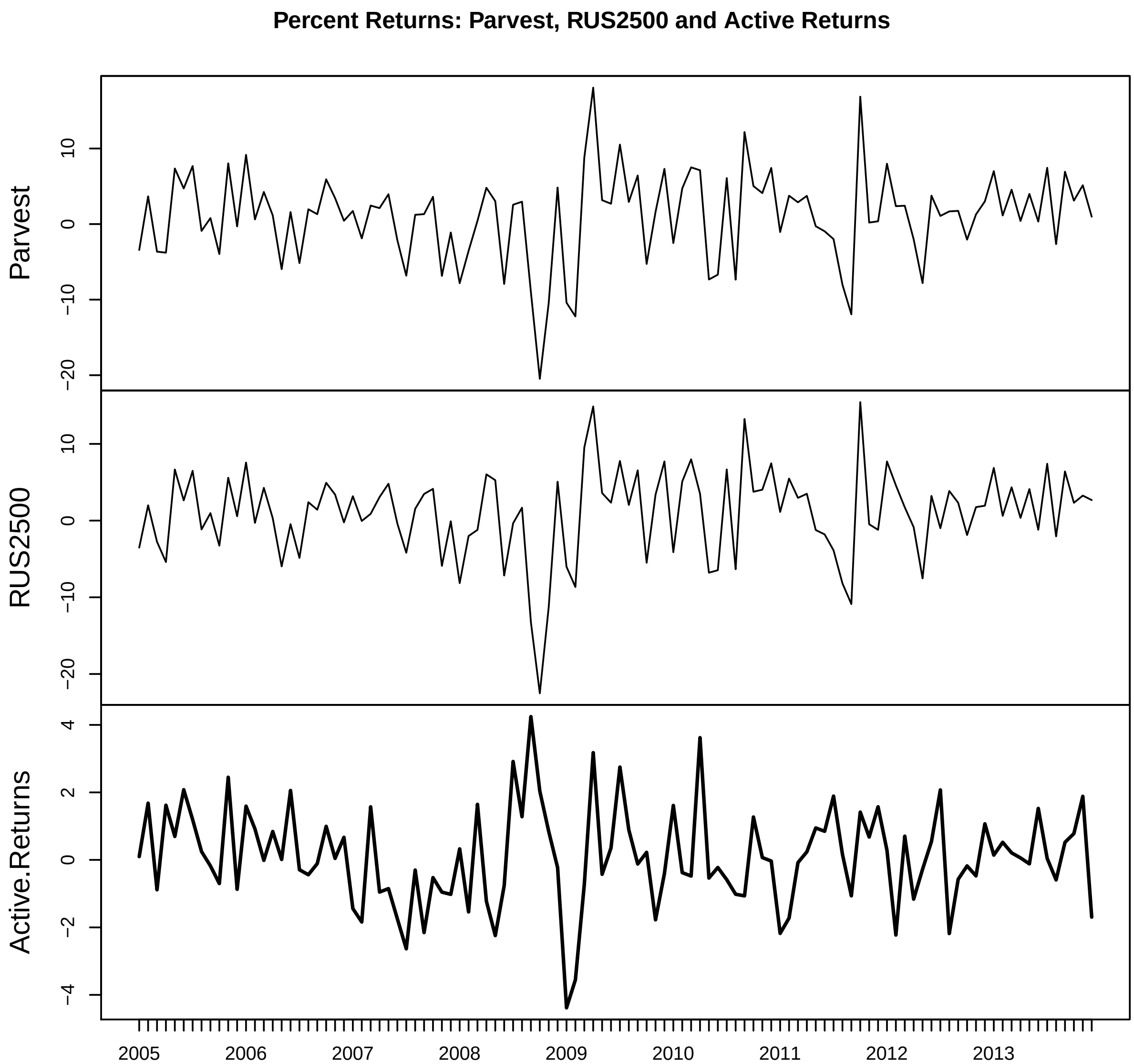
The above Cusum methodology for monitory active managers is implemented in the R function `cusumActMgr`. The following illustrates its use.

First, load the example data set `cusumData` which contains two time series of monthly returns: `Parvest` and `RUS2500` (Russell 2500 data). `Parvest` is a single stock proxy for for an actively managed portfolio, and `RUS2500` is its benchmark.

```
data(cusumData)
head(cusumData)

##              Parvest     RUS2500
## Jan 2005 -0.03428022 -0.03525696
## Feb 2005  0.03669338  0.01987962
## Mar 2005 -0.03649360 -0.02763560
## Apr 2005 -0.03783733 -0.05404117
## May 2005  0.07355658  0.06661316
## Jun 2005  0.04716800  0.02637423

active.ret = cusumData[,1] - cusumData[,2]
names(active.ret) = "Active Returns"
alldat = cbind(cusumData,active.ret)
plot.zoo(100*alldat, xlab = "", lwd = c(1,1,2),
       main = "Percent Returns: Parvest, RUS2500 and Active Returns")
```



The following code reveals the arguments to the function `cusumActMgr`:

```
args(cusumActMgr)

## function (portfolioName, benchmarkName, data, upperIR = 0.5,
##     lowerIR = 0, lambda_in = 0.1, lambda_out = 0.2, huberize = 4,
##     filterStd = FALSE)
## NULL
```

For a detailed description of these arguments, see the help file with:

```r
help(cusumActMgr)
```

Next, use `cusumActMgr` to compute the results needed to analyze the performance of this active manager:

```r
results = cusumActMgr(portfolioName = "Parvest", benchmarkName = "RUS2500",
                      data = cusumData)
```

The object `results` is a `list` with components that have the following component names:

```r
names(results)

##  [1] "Logarithmic_Excess_Returns" "Annual_Moving_Average"
##  [3] "Tracking_Error"             "Information_Ratios"
##  [5] "Lindley's_Recursion"        "Annualized_Cusum_IR"
##  [7] "Annualized_Cusum_ER"        "Means"
##  [9] "Protractor_IR"              "Protractor_ER"
## [11] "Standard_Deviations"        "Excess_Volatility"
## [13] "AIR"                        "AER"
```

These components are mostly `xts` objects.

Print and Summary Methods

** PRINT AND SUMMARY METHODS TO BE ADDED IN NEXT RELEASE **

Active Manager Charts

Running the program generates a set of eight graphs which display the following quantities in an appropriate format on two pages:

1. Monthly active returns
2. Annualized robust EWMA tracking error estimated using Equations (3) and (4),
3. Monthly estimate of the current Information Ratio (Annualized),

4. Excess volatility (defined to be difference between the robust EWMA volatility estimates for the portfolio and its benchmark),

5. Cusum plot for active returns, along with a protractor of slopes to allow the user to make a visual estimate of the performance of the portfolio relative to its benchmark,

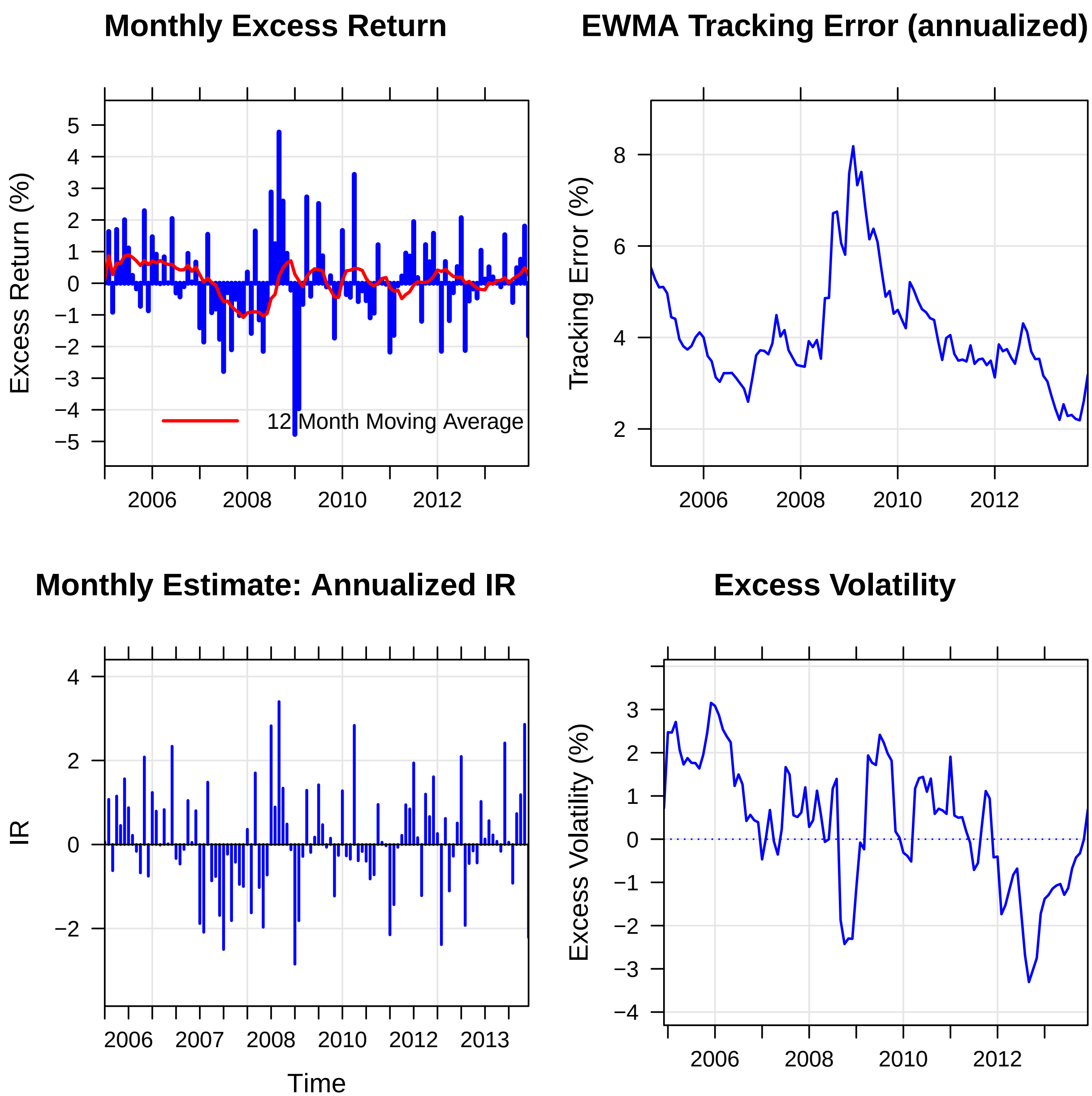
6. Cusum plot for tjhe Information Ratio, also with a protractor of slopes to allow the user to make a visual estimate of the Information Ratio of the portfolio,

7. Page's plot, which displays the scaled likelihood ratio described by (6)

8. A robust regression fit of Parvest returns to the RUS2500 returns, along with an the slope (β) and scaled intercept (annualized α)

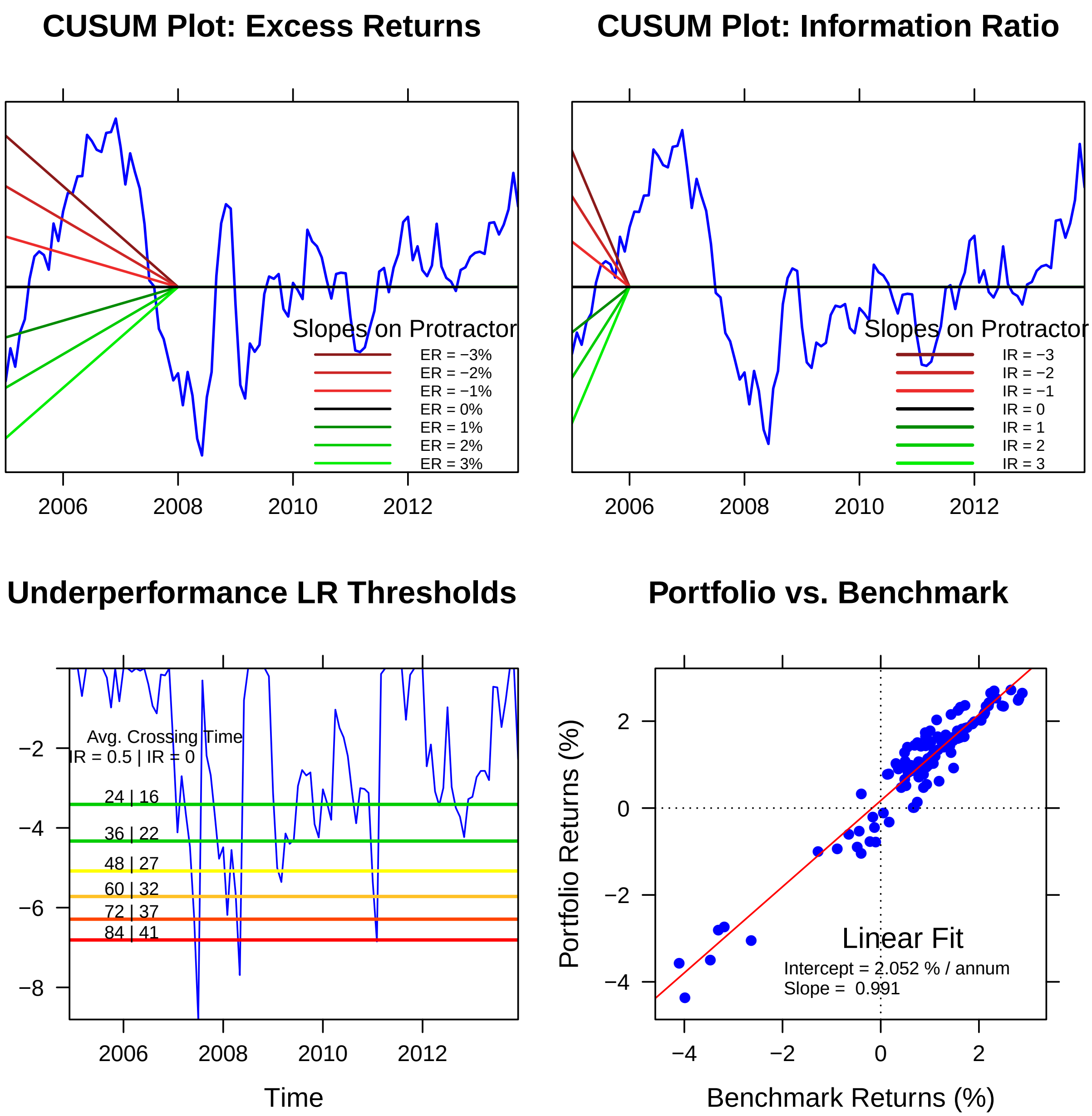
The first four plots are generated on a single page using the command

```
chartCusum(results, which = 1)
```



while the next four plots are generated using the command

```
chartCusum(results, which = 2)
```



Finally, all eight graphs can be generated on two pages using the command:

```r
chartCusum(results, which = c(1,2))
```

The seventh plot (Page's Procedure: Likelihood Ratio Test) is the beating heart of the Cusum procedure, and displays the results of (6). It has seven horizontal lines, each of which corresponds to one of the lines in Table 1 (the difference between the levels displayed on the y-axis of the plot and the thresholds in Table 1 are entirely the result of the quantities in the table being expressed in annualized, i.e., scaled by $\sqrt{12}$, terms). Each horizontal line has two numbers associated with it: the expected times to cross that threshold conditioned on the Information Ratio being 0.5 and 0 (columns 2 and 3 of Table 1). The bottom most line corresponds to the last line of the table, and when the likelihood ratio crosses this line, an alarm is raised, triggering an investigation.

If, after the investigation is completed, the alarm is found to be a false alarm, the plot is reset to 0 and monitoring is restarted. On the other hand, if the investment process is found to be deficient, the plot can be stopped if the portfolio is shut down, or reset after the required changes have been made. The Cusum procedure will automatically adapt to the new level of tracking error, and monitoring will continue as before.