Financial Journalism with R

Bill Alpert

R proved itself a sharp tool in testing the stock picks of Jim Cramer, a popular US financial journalist. We examined Cramer's advice for a recent cover story in *Barron's*, the Dow Jones & Co. stock market weekly, where I am a reporter and floundering R user (Alpert, 2007). The August 20, 2007 story should be accessible without subscription at the *Barron's* website (`http://www.barrons.com`).

The 52-year-old Cramer once ran a hedge fund which racked up 24% annualized returns over about a dozen years. His current celebrity comes from the *Mad Money* television show on the cable network CNBC, in which he makes Buy and Sell recommendations to the accompaniment of wacky sound effects and clownish antics. A few have attempted to measure the performance of modest samples of Cramer's picks (Engelberg et al., 2007; Nayda, 2006). Yet Cramer makes almost 7,000 recommendations a year, according to the count of a database at his *Mad Money* website (`http://madmoney.thestreet.com/`). He has never reported the performance of all those stock picks. I figured I'd try.

As in most projects, data collection was the hard part. I called Cramer and asked for any records of his *Mad Money* picks. After ripping into me and all journalists who've reviewed his show, he stopped taking my calls. Meanwhile, I found a website maintained by a retired stock analyst, who had tallied about 1,300 of Cramer's *Mad Money* recommendations over two years. I also found the abovementioned database at Cramer's official website, which recorded over 3,400 recommendations from the preceding six months. This Cramer site classified his picks in ways useful for subsetting in R, but conspicuously lacked any performance summaries. I turned these Web records of his stock picks into Excel spreadsheets. Then I downloaded stock price histories from Edgar Online's I-Metrix service (`http://I-Metrix.Edgar-Online.com`), using some Excel macros. None of this was trivial work, because I wanted a year's worth of stock prices around each recommendation and the date ranges varied over the thousands of stocks. Financial data suppliers can deliver a wealth of information for a single, common date range, but an "event study" like mine involved hundreds of date ranges for thousands of stocks. Most finance researchers deal with this hassle by using SAS and a third-party add-on called Eventus that eases data collection. But I wanted to use R.

I reached out to quantitative finance programmer Pat Burns and Pat wrote some R code for our event study style analysis. Pat has posted his own working paper on our study at his website (`http://www.burns-stat.com`). R's flexibility was useful because we needed a novel data structure for the Cramer analysis. In most analyses, the data for prices (or for returns) are in a matrix where each column is a different stock and each row is a specific date. In our case, each stock recommendation had the same number of data points, so a matrix was a logical choice. However, instead of each row being a specific date, it was a specific offset from the recommendation date. We still needed the actual date, though, in order to get the difference in return between the stocks and the S&P 500 on each day – to see if Cramer's picks "beat the market." Pat's solution was to have a matrix of dates as a companion to the matrix of prices. It was then a trivial subscripting exercise to get a matrix of S&P returns that matched the matrix of returns for the recommendations. Many stocks were recommended multiple times, so the column names of the matrices were not unique.

Once the data were in R, it was fairly easy to test various investment strategies, such as buying the day after Cramer's 6 pm show or, instead, waiting two days before buying. Any investment strategy that actively picks stocks should at least beat the returns you'd get from a passive market-mimicking portfolio with comparable risk (that is, similar return variance) – and it should beat the market by enough to cover trading costs. Otherwise you ought to keep your money in an index fund. You can see the importance of market-adjusting Cramer's returns by comparing the red and blue lines in Figure 1. The Nasdaq Composite Index is arguably a better match to the riskiness of Cramer's widely-varying returns. We made his performance look better when we used the S&P.

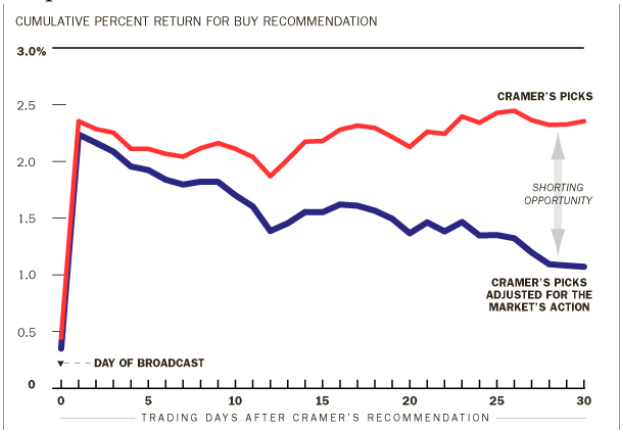

Figure 1: Average cumulative percentage log return from the day of broadcast for approx. 1,100 Mad Money recommendations recorded at `http://www.yourmoneywatch.com`. Blue line shows return relative to S&P 500 Index.

The red line shows his average pick's unadjusted log return, while the blue shows the log re-

turn relative to the Standard & Poors 500 Index. The data were roughly 1,100 Buy recommendations over the period from July 2005 to July 2007. The lines' lefthand peaks mark the trading day after each evening broadcast, when enthusiastic fans have bid up Cramer's pick. We ultimately tested several dozen investment strategies.

The results were disappointing for someone who wants to follow Cramer's advice. You could not beat the market by a statistically significant amount if you followed his recommendations in any way readily available to a viewer. But we did find that you might make an interesting return if you went *against* Cramer's recommendations – shorting his Buys the morning after his shows, while putting on offsetting S&P 500 positions. This shorting opportunity appears in Figure 1, as the widening difference between the red and blue lines. If a viewer shorted only Cramer's recommendations that jumped 2% or more on the day after his broadcasts, that difference could earn the viewer an annualized 12% on average (less trading costs). The bootstrapped 95% confidence intervals of this difference ranged from 3% to 21%. (For background on bootstrap techniques, see Efron and Tibshirani, 1993)

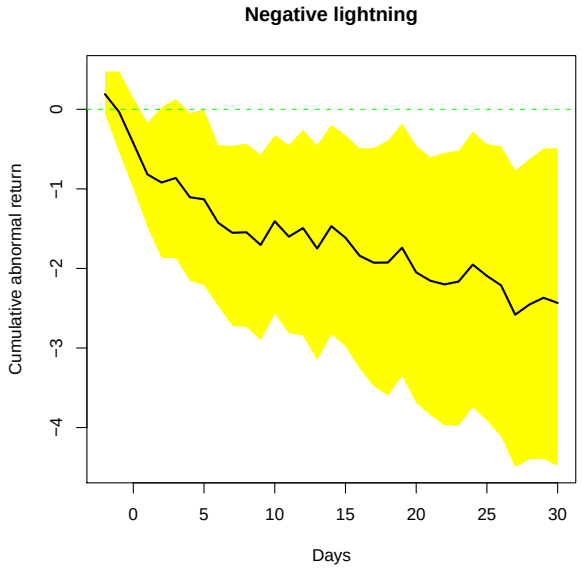

Figure 2: Cumulative log return relative to S&P 500 on about 480 of Cramer's off-the-cuff Lightning Round Sell recommendations, as recorded at madmoney.thestreet.com. The yellow band is a bootstrapped 95% confidence interval, showing that the negative return – desirable for Sell recommendations – is clearly different from zero

One reason we tested so many different strategies is that when I finally started getting responses from Cramer and CNBC, they kept changing their story. They argued about selection. Neither of my databases were reliable records of Cramer's picks, CNBC said, not even the database endorsed by Cramer at his website. Instead of giving me a record of his two-years' of stock picks, they suggested that I watch tapes of all his shows, which they'd gladly provide me. It reminded me of the endless tasks assigned in fairy tales – separating peas from the ash pile.

They also argued that certain subsets of Cramer's recommendations were the only proper ones to count . . . as if viewers would somehow know which of his screaming "Buys" he meant them to ignore. As it happened, the segment of his show that Cramer argued most shrilly for us to deselect (the "Lightning Round" with his impromptu responses to phoned-in questions) was the only segment with any statistically-significant marketbeating returns – in this case, on the Sell recommendations (see Figure 2). Embarrassingly, the Sell recommendations that Cramer prepared in advance went up (see Figure 3).

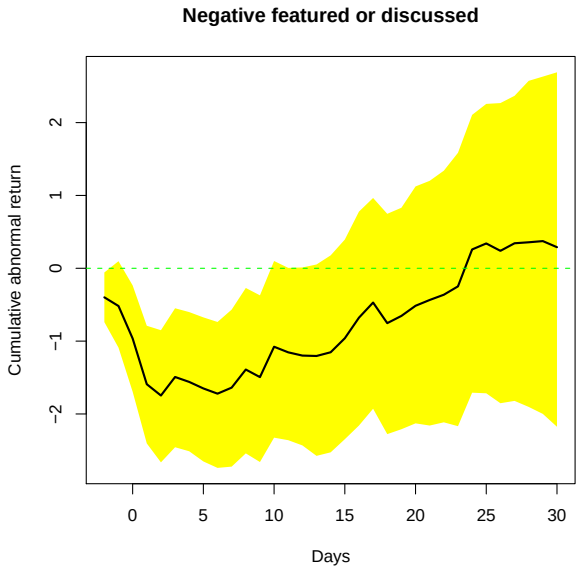

Figure 3: Cumulative log return relative to S&P 500 on about 160 of Cramer's Prepared Sell recommendations as recorded at madmoney.thestreet.com

CNBC and Cramer also kept shifting their claim about when Cramer advises his audience to act on his picks. After I told Cramer that viewers got market-lagging returns if they bought the day after his recommendations (when most viewers clearly do their buying), he angrily said that he "always" advises waiting until the second day. When I came back with evidence that a second-day purchase also lagged the market (as is clear in Figure 1), CNBC said that Cramer's advice had "always" been to wait a week or two after each show. Later still, they said that a good test should wait exactly five days.

Such data dredging and methodological debates are probably not unfamiliar to readers of this newsletter who use R for high stakes evaluations of pharmaceuticals or government interventions, but journalism may be extreme in its extensive interac-

tion with the subjects of its analyses. Academics rarely contact authors they critique – indeed, post-publication data repositories aim to make it easy to replicate a study without questioning the authors. Good journalists do the opposite: we contact the subjects of our investigations early and often.

And of course, R let us flexibly change our analysis every time CNBC and Cramer changed their story. When they argued for a different selection, R made it easy to create a new data subset. When they argued for a different holding strategy, R's indexing facility let us start and stop our analysis on different dates. In fact, when I begged CNBC for their own analysis of Cramer's performance, they said something that should warm the hearts of all you folks who've made R the powerful environment it is. CNBC told me not to expect a timely response from them because it was obvious that Pat and I had spent *months* on our analysis. In truth, Pat put in less than a week's work.

Acknowledgments

Essential services for this project included: R coding by Pat Burns, of Burns Statistics in London; Excel macros by Edgar Online analyst Elias-John Kies and from Tufts University undergraduate Madison Mcgaffin; wise statistical pointers from Tim Hesterberg, of Insightful Corp. in Seattle . . . none of whom bear responsibility for any woebegone errors.

Bibliography

B. Alpert. Shorting Cramer. *Barron's*, 2007.

B. Efron and R. Tibshirani. *Introduction to the Bootstrap*. Chapman and Hall, 1993.

J. Engelberg, C. Sasseville, and J. Williams. Attention and Asset Prices: the Case of Mad Money. Technical report, Kellogg School of Management, 2007. `http://papers.ssrn.com`.

N. Nayda. You! Me! Let's Get Ready to Make Mad Money!!! Technical report, Elmont Virginia Elementary School, 2006. Science fair project.

Bill Alpert
Barron's, Dow Jones & Co., U.S.A.
`william.alpert@barrons.com`

Need A Hint?

Sanford Weisberg and Hadley Wickham

Suppose you have created an object in R, for example from a regression fit using `lm` or `loess`. You know that auxiliary functions exist that do useful computations on the object, but you can't remember their names. You need a hint on what to do next.

The `hints` function in the **hints** package does just this, finding a list of appropriate functions to jog your memory. For example, Figure 1 shows a list of hints for a `lm` object.

The output lists methods for generic functions like `print` specific to the class you specify, as well as searching the documentation to find all mentions of the class. You can then use the usual help mechanism to learn more about each of these methods and functions.

The `hints` function has three arguments:

```
hints(x, class=class(x), all.packages=FALSE)
```

If specified, the argument `x` can be any R object. For example, `x` might have been created by `x <- lm(y z)`. `hints` determines the S3 class of the object, and then looks for functions that operate on that class. The S3 class of an object is a character vector, and may consist of multiple strings, as, for example, a generalized linear model which has class `c("glm", "lm")`. If `x` is not given, then you can specify the `class` you want hints about as character vector.

The hints function will look for methods and functions in all currently loaded packages. For example, the hints for `lm` would be different if either the **car** or the **alr3** packages have been loaded, since both of these add methods and functions for `lm` objects. Similarly, `hints(class="lda")` would return methods only if the package **MASS** were loaded, since all the relevant methods and functions are in that package. You can get hints for all your packages by setting `all.packages=TRUE`, but note that this works by `require`'ing all available packages so may be time consuming.

The **hints** package also includes an `xtable` method so, for example, `xtable(hints(m1))` would have produced a version of Figure 1, but in LaTeX format.

The function isn't foolproof, as it depends on the quality of documentation written by others. It may find irrelevant functions if the name of the class appears in the documentation for the irrelevant function. It can miss functions, too. For example, the function `coeftest` in the **lmtest** package can be used with `lm` objects by applying the function `coeftest.default`.