

# Simulation of Leveraged ETF Volatility Using Nonparametric Density Estimation

Matthew Ginley, David W. Scott, Katherine E. Ensor

Department of Statistics, Rice University, Houston, TX, USA

Email: [mginley@rice.edu](mailto:mginley@rice.edu), [scottdw@rice.edu](mailto:scottdw@rice.edu), [ensor@rice.edu](mailto:ensor@rice.edu)

Received 30 September 2015; accepted 27 November 2015; published 30 November 2015

Copyright © 2015 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

---

## Abstract

Leveraged Exchange Traded Funds (LETFs) are constructed to provide the indicated leverage multiple of the daily total return on an underlying index. LETFs may perform as expected on a daily basis; however, fund issuers state that there is no guarantee of achieving the multiple of the index return over longer time horizons. LETF returns are extremely volatile and funds frequently underperform their target for horizons greater than one month. In this paper, we contribute two nonparametric simulation methods for analyzing LETF return volatility and how this is related to the underlying index. First, to overcome the limited history of LETF returns data, we propose a method for simulating implied LETF tracking errors while still accounting for their dependence on underlying index returns. This allows for the incorporation of the complete history of index returns in an LETF returns model. Second, to isolate the effects of daily, leveraged compounding on LETF volatility, we propose an innovative method for simulating daily index returns with a chosen constraint on the multi-day period return. By controlling for the performance of the underlying index, the range of volatilities observed in a simulated sample can be attributed to compounding with leverage and the presence of tracking errors. Our nonparametric methods are flexible-easily incorporating any chosen number of days, leverage ratios, or period return constraints, and can be used in combination or separately to model any quantity of interest derived from daily LETF returns.

## Keywords

Nonparametric Density Estimation, Exchange Traded Fund, Realized Volatility

---

## 1. Introduction

The class of securities commonly known as LETFs, or Leveraged Exchange Traded Funds, began trading on the

New York Stock Exchange in 2006. LETFs are managed by their issuers with a mandate to provide a multiple of the daily return of the underlying asset, and are developed to provide professional investors a convenient means for the short term hedging of market risk. Since their issuance, their popularity in the investment world has risen steadily and now there are investors promoting buy and hold strategies that feature long term positions in LETFs [1] [2]. The lack of understanding of the risks associated with these strategies and more generally LETF return dynamics was a primary motivation for our work. In this paper, our objective is to offer improved tools for analyzing the risks involved with investing in LETFs.

The first Exchange Traded Fund (ETF) began trading in 1993. By 2015, there are over 1400 ETFs from over 15 different issuers, with over \$1 trillion invested in ETFs globally [3]. The existing universe of ETFs covers all major asset classes such as equities, debt, commodities, and real estate. Many ETFs are referenced as the underlying asset of actively traded options. We will focus on a specific subset of ETFs called Leveraged ETFs (LETFs) which feature a leverage ratio different than 1:1 as part of their fund structure.

The mandate for LETFs is to provide the daily return of an underlying asset, most often an equity index, at the multiple indicated by the leverage ratio. As of 2015 there are fund offerings featuring leverage ratios of 2:1, -2:1, 3:1, and -3:1. Going forward, we will simply refer to leverage by its multiple (*i.e.* 2, -2, 3, -3). For example, if the underlying index returned 1% on the day, then LETFs with leverage multiples of 2, -2, 3, and -3 would have a mandate to return 2%, -2%, 3%, and -3% on the day, respectively. Positive values indicate long market exposure, while negative multiples indicate short exposure, or betting that the market will decline. Funds that feature negative leverage multiples are often times referred to as Inverse ETFs, but we will not make this distinction because the returns dynamics are the same except for the obvious negative sign on the leverage term.

In addition to the included leverage, providing investors with the ability to easily acquire short exposure is an important selling point for LETF issuers. Before these LETFs started trading in 2006, leveraging one's assets or betting against the market required utilizing expensive margin and collateral accounts or navigating the challenges involved with trading options [4]. Investors have since grown to appreciate the conveniences offered by the LETF structure and as in 2015 there are over 100 funds with a total notional value over \$20 billion listed on exchanges worldwide, with Direxion, ProShares, Credit Suisse, and Barclays issuing the majority of them [5].

Not long after LETFs started trading, researchers began to examine the leveraged returns they provide and the associated daily compounding. Jarrow [6] and Avellaneda and Zhang [7] were some of the first to do so. The work by both sets of authors is very similar. They clearly prove that the multi-day return on an LETF is not the same as the return on an (unleveraged) ETF with the same underlying index and relate the difference to the ETF's volatility and borrowing costs. Avellaneda and Zhang also provide a strong collection of empirical results to reinforce their proof. In a similar fashion, Lu, Wang, and Zhang [8] provide empirical results that extensively quantify the underperformance of LETFs with respect to their benchmark over multi-day periods. They claim generally that with a leverage multiple of 2, or we say "2x leverage", one can expect an LETF to deliver its targeted return for periods up to a month, and then they quantify expected deviations for longer horizons.

The work by Dobi and Avellaneda [9] takes the discussion of LETFs a step further. They highlight the total return swap exposures that are rebalanced daily by LETF issuers to provide the leveraged return target as a cause of LETF underperformance. We strongly agree with their comment that LETF underperformance "... is especially true for longer holding periods and for periods of high volatility". Tuzun [10] expands on the ramifications of the swap exposures that are held by the LETF issuers. He notes the similarities between daily rebalancing necessitated by LETFs and portfolio insurance strategies from the 1980s. This work is written from the perspective of market regulators, but we found it helpful because it emphasized the effects of consecutive gains or consecutive losses on LETF performance. With LETFs a period of consecutive gains produces leveraged gains on leveraged gains, resulting in a compound return that is much greater than would be experienced without leverage. This scenario should not be difficult to understand, but in the case of consecutive losses with leverage, the authors demonstrate that this also results in improved performance (relatively). In other words, relative to the same index exposure without leverage, the magnitude of the cumulative loss experienced with leverage is less. This is because with every consecutive negative return, exposure is decreased before the next loss, and with leverage the decrease of exposure is greater.

Cooper [11] provides a detailed review of volatility drag and LETF dynamics in an attempt to prove that there exists a leverage ratio greater than one that is optimal for buy and hold investing in any equity index. We are doubtful of his claims because of the assumption of no tracking errors. We show later this would be poor

judgment with the LETFs that we analyze. Also, Bouchev and Nemtchinov [12] show that underperformance of high volatility stocks compared to low volatility stocks is mostly the result of volatility drag. There is a parallel that could be considered between their thesis and the performance of LETFs with high volatility underlying indexes compared to LETFs with low volatility underlying indexes, but we do not investigate it here.

Qualitatively speaking, we agree with the above mentioned authors that LETFs frequently underperform over time horizons greater than one month, underperformance is greater in times of high volatility, and that underperformance is intrinsically related to volatility drag. But, we view our contributions as an empirical framework that complements their analytical models and we do not attempt to reproduce the results on which their conclusions are based. Instead, our intention is to provide methods that can help LETF investors better account for these dynamics. The remainder of this paper is organized as follows.

In Section 2 we review the LETF returns dataset used throughout this paper, present a visual motivation for our contributions, and describe two statistics for measuring the realized volatility of LETF returns-sample standard deviation and Shortfall from Maximum Convexity (SMC). SMC is a variant of volatility drag that specifically accounts for the effects of leverage. We defined SMC and documented the advantages of using it to measure realized volatility in a separate publication [13], which is summarized in Section 2.

In Section 3 we propose a method for simulating LETF tracking errors. The proposed method allows us to produce simulated LETF returns derived from index returns and LETF tracking errors that are simulated separately. In turn, we can make use of the entirety of their respective datasets, greatly improving the robustness of our density estimates used in simulation.

In Section 4 we exploit this separation by proposing a method for simulating index returns with a constrained total period return. Constraining the underlying index performance allows for greater nuance with modeling LETF volatility. With this method, a practitioner can formulate more customized LETF volatility summaries by including a specific returns forecast for the underlying index.

Finally, we demonstrate how to combine the two simulation methods with a brief case study in Section 5.

## 2. Visualizing LETF Returns and Measuring Realized Volatility

We have selected a wide ranging set of 10 equity indexes referenced as the underlying asset by 10 pairs of long and short LETFs for demonstrating our work (10 funds with a +3 leverage multiple and 10 with a -3 multiple). All of the included pairs of  $\pm 3x$  levered ETFs are issued by Direxion and ProShares. Our set includes established U.S. market indexes (S&P 500, Dow Jones Industrial Average, Russell 2000, NASDAQ 100), sector specific indexes (NYSE ARCA Gold Miners, Dow Jones U.S. Financials, S&P Technology Select), and international equity indexes (MSCI Developed Markets, MSCI Emerging Markets, and Market Vectors Russia). The specific instruments included in the demonstration set are listed in the Appendix in [Chart 1](#) and [Chart 2](#).

Throughout this paper, we assume that daily LETF returns (for a generic ticker *LEFT*) and their matching daily index returns (generic ticker *Index*) are related through the following decomposition:

$$R_{LEFT,t} = \beta_{LEFT} R_{Index,t} - \text{MgmtFee}_{LEFT} + \epsilon_{LEFT,t} \quad (1)$$

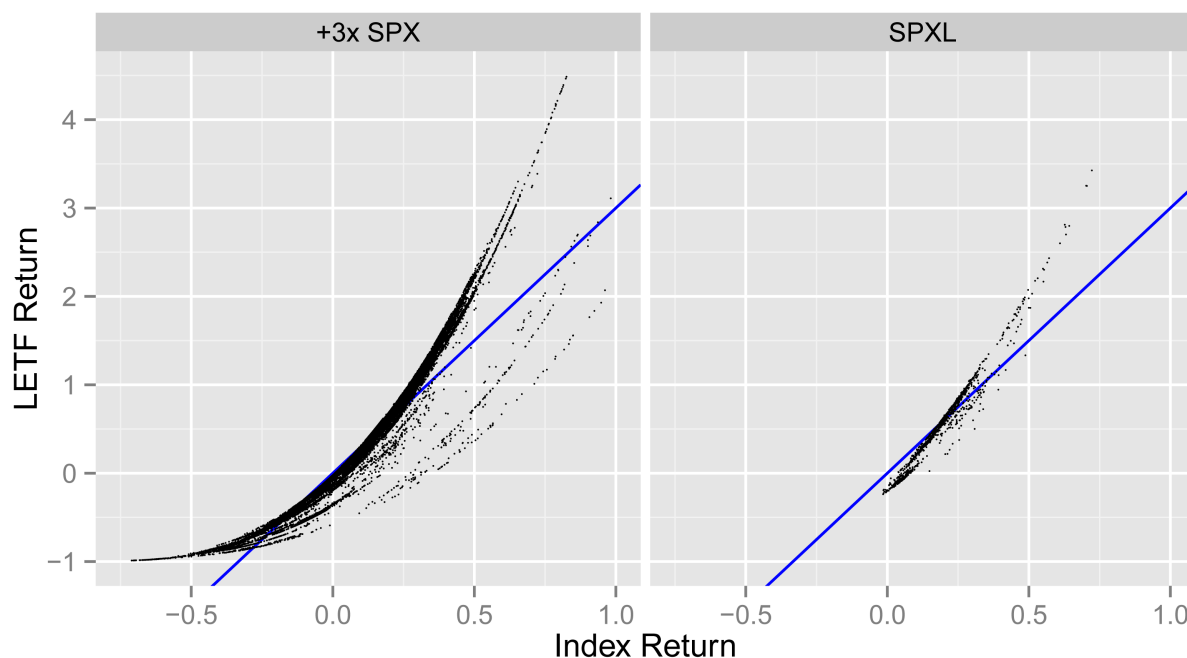
where  $R_{LEFT,t}$  is the daily return on day  $t$  for fund ticker *LEFT*,  $R_{Index,t}$  is the daily return on day  $t$  for the underlying ticker *Index*,  $\beta_{LEFT}$  is the indicated leverage multiple for *LEFT* (constant),  $\text{MgmtFee}_{LEFT}$  is the indicated rate of daily management expenses for *LEFT* (constant), and  $\epsilon_{LEFT,t}$  is the tracking error on day  $t$  for *LEFT* (random variable).

The leverage multiple and annualized management expenses are indicated in a given LETF's fund prospectus. As mentioned, since 2015 there are commercially available LETFs with leverage multiples of 2, -2, 3, and -3. All of the management fees indicated on the LETFs in our demonstration set were listed as 0.95% annually, or a daily rate of approximately 0.375 basis points [14]-[17]. Fees for other LETFs from Direxion and ProShares or other fund issuers are the same or comparable [5].

All the daily returns data for the 10 indexes and 20 LETFs in the demonstration set are sourced from Bloomberg and the LETF leverage and fees constants are sourced in their fund prospectuses provided by their respective issuers [16]-[18]. No other sources of data are referenced in this paper.

### 2.1. Visualizing LETF Returns

Our contributions can best be motivated through visualization, and so we begin with [Figure 1](#). There are two



**Figure 1.** Scatter plots of 252 day returns for hypothetical LETF “+3x SPX” and SPXL.

scatter plots of 252 day compound LETF returns (y axis) against the matching 252 day compound index returns (x axis) from the same periods. The right plot is for +3x LETF Ticker SPXL, the left plot is for the hypothetical LETF “+3x SPX” created by daily, leveraged compounding of S&P 500 returns, and then for both the underlying index is the S&P 500. The diagonal blue line in each plot corresponds to  $y = 3x$ , and is provided to indicate what return would be achieved by a leveraged investment in the index without daily compounding. It has been said time and again that “a picture is worth 1000 words,” but in our case we could say 10,000 words, the approximate length of this paper.

Three extraordinarily important observations need to be made. First, there are many more points representing a much wider range of outcomes in the left plot, which is derived from the entire daily returns series of the S&P 500 (21,591 intervals of 252 days), compared to the right plot that includes only the SPXL returns series (1288 intervals of 252 days). Second, there is an interesting crescent shape with a hard upper bound apparent in the left plot, where the overwhelming majority of points are clustered near the upper bound. Third, again in the left plot, points seem to disperse from the upper bound in a way that maintains some degree of curvature such that the range of dispersion is not constant. These characteristics were readily observable when the same plot was generated using other LETF and index pairs.

**Figure 1** illustrates the advantages of having more data. The right plot was limited to the returns data available for SPXL, which extends back to 2008. On the other hand, the left plot used all the data available for the S&P 500, which extends back to the 1920s. Except for the Market Vectors Russia Index (MVRSX) there are at least 16 years of data for every index that we consider in this paper. Given the dependence of an LETF’s daily returns on the matching returns of the underlying index, it follows naturally that we should develop schemes for modeling LETF returns that can incorporate and control for the variation in the complete history of index returns data. Our methods presented in Sections 3 and 4 address these two motivations, respectively.

The potential enhancements offered by our methods are especially significant given the fact that North American equity indexes have generally been in an upward trend since March 2009. In other words, the majority of available LETF returns were observed when the index returns that they are derived from were positive or compound to a positive value over periods of months or years. This can be seen in **Figure 1**, where the right plot has no points with an  $x$  value less than 0 (the underlying index return), but there are many points that satisfy this condition in the left plot. Obviously, any quantitative LETF analysis would benefit from the ability to include data from downward trending markets and isolate the resulting effects.

## 2.2. Measuring Realized Volatility

Modeling the volatility of asset returns as the standard deviation of the returns distribution has been commonly accepted since the publication of Markowitz' benchmark paper on portfolio theory [19]. Accordingly, sample standard deviation would be the usual measure of realized volatility of a  $p$  dimensional vector of daily log LETF returns  $\log(1 + \mathbf{R}_{LETF})$ . We denote this statistic as PSD, or Periodized Standard Deviation.

$$\text{PSD}(\mathbf{R}_{LETF}) = \sqrt{p} * \left[ \frac{1}{p} \sum_{j=1}^p \left( \log(1 + R_{LETF,j}) - \log(1 + \bar{R}_{LETF}) \right)^2 \right]^{\frac{1}{2}} \quad (2)$$

$$= \left[ \sum_{j=1}^p \left( \log(1 + R_{LETF,j}) - \log(1 + \bar{R}_{LETF}) \right)^2 \right]^{\frac{1}{2}} \quad (3)$$

$$\bar{R}_{LETF} = \prod_{j=1}^p (1 + R_{LETF,j})^{\frac{1}{p}} - 1 \quad (4)$$

where  $\bar{R}_{LETF}$  is the geometric mean of the elements of the geometric returns vector  $\mathbf{R}_{LETF}$ .

Alternatively, we will measure realized volatility with Shortfall from Maximum Convexity (SMC), a novel statistic that was designed specifically for application to LETF returns [13].

$$\text{SMC}(\mathbf{R}_{LETF}, \mathbf{R}_{Index} | \beta_{LETF}) = \exp \left[ \sum_{j=1}^p \left( \log(1 + \beta_{LETF} \bar{R}_{Index}) - \log(1 + R_{LETF,j}) \right) \right] - 1 \quad (5)$$

$$\bar{R}_{Index} = \prod_{j=1}^p (1 + R_{Index,j})^{\frac{1}{p}} - 1 \quad (6)$$

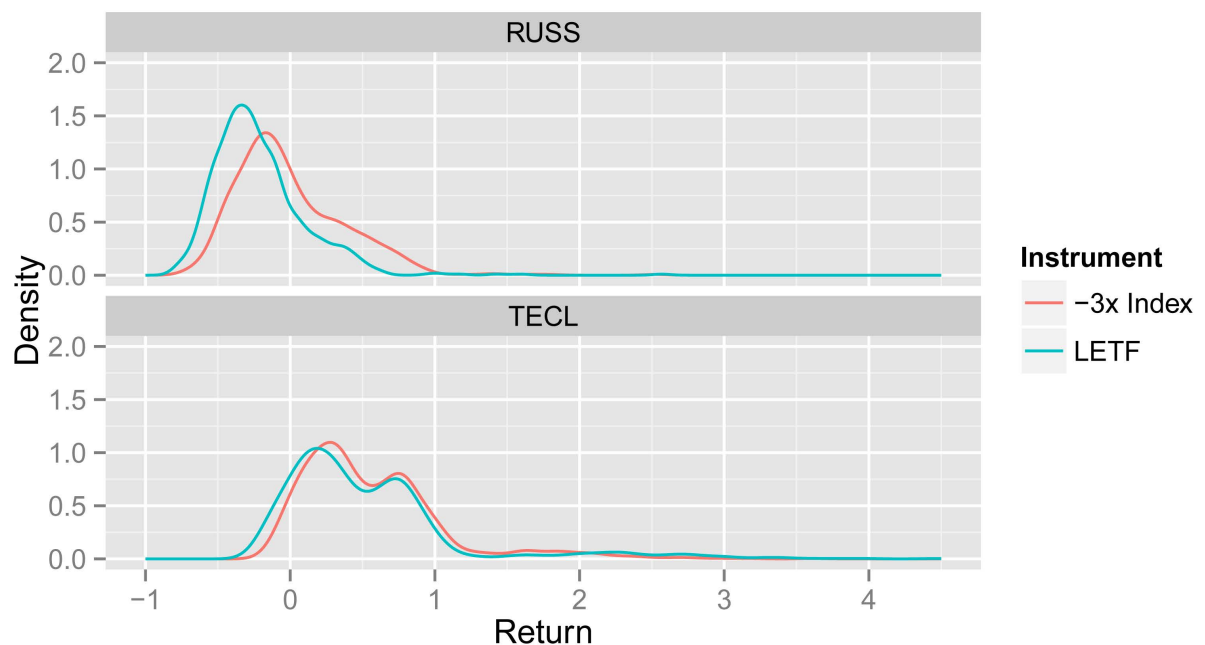
where  $\beta_{LETF}$  is the indicated leverage multiple for LETF (constant),  $\mathbf{R}_{Index}$  is the matching  $p$  dimensional vector of daily returns for Index, and  $\bar{R}_{Index}$  is the geometric mean of the elements of  $\mathbf{R}_{Index}$ . What differentiates SMC from PSD is subtle—a mean return of  $\beta_{LETF} \bar{R}_{Index}$  is used instead of  $\bar{R}_{LETF}$ , and the sum of daily differences is exponentiated instead of taking the square root of the sum of squared, daily differences.

These changes were inspired by the geometric details of the plotting arrangements used in **Figure 1**. SMC is simply the vertical distance between a data point and the point on the convex upper bound that is located at the same  $x$  value (calculating the exact form of points on the upper bound is trivial). The interpretation of this measure is the geometric excess return of a hypothetical, maximum return derived from the index return, with respect to the LETF return. We use SMC because the interpretation of PSD becomes problematic with non-normal, dependent data, and when applied to LETF returns SMC provides more information, statistically speaking. These arguments are covered in much greater detail within the original proposal for SMC, and we refer interested readers there for more information.

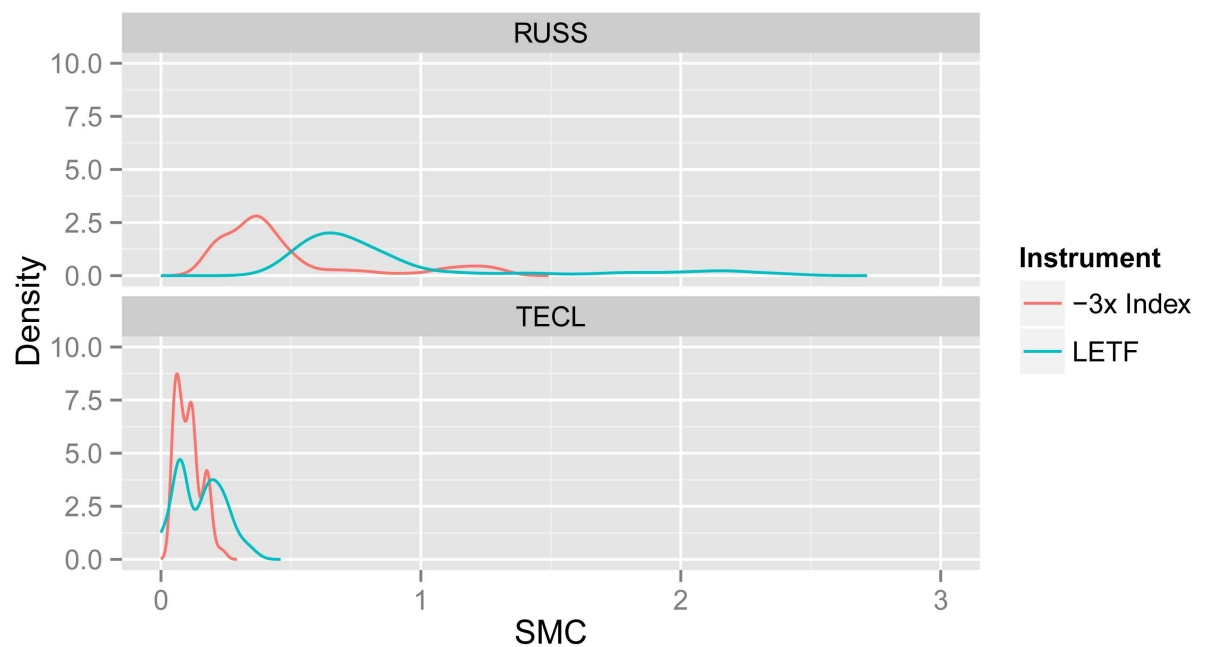
## 3. Simulation of Tracking Errors Using Nonparametric Density Estimation

A naive approach for modeling LETF returns would be to simply assume no tracking errors and approximate daily returns as the underlying daily index return with leverage and net of management fees. These approximated returns, after compounding, could be used in the denominator when calculating SMC in place of a true LETF return. This naive approach is simple and allows for the incorporation of the entire history of index returns data; however, it ignores the impact of LETF tracking errors on returns and realized volatilities.

This impact becomes significant over time, as demonstrated in **Figure 2** and **Figure 3** for two LETFs. First, density curves of 252 day returns of LETF Ticker RUSS (top) and LETF Ticker TECL (bottom) are plotted against 252 day returns of their respective underlying indexes, but compounded daily with a leverage multiple of  $-3$ . Note the separation between the density curves, which is due to the cumulative effect of tracking errors because the index returns were calculated with daily compounding and leverage. Second, density curves of 252 day SMC are plotted using the same data and plotting arrangements. Again, note the separation between the density curves, and that the separation is much greater when visualizing realized volatility compared to visualizing returns.



**Figure 2.** Densities of 252 day returns for LETF Tickers RUSS and TECL and their respective indexes compounded daily with  $-3\times$  leverage.



**Figure 3.** Densities of 252 Day SMC for LETF Tickers RUSS and TECL and their respective indexes compounded daily with  $-3\times$  leverage.

In light of this evidence, it is our opinion that approximating LETF returns by assuming no tracking errors would provide for inaccurate analyses, and in response we propose the nonparametric method for simulating LETF tracking errors described in this section. Our method estimates filtering and smoothing densities using only data from the days when LETFs were trading. The proposed method effectively allows for simulating LETF returns derived from index returns and LETF tracking errors that are simulated separately, thus incorporating the entire history of index returns and LETF returns data in an extremely efficient manner.



### 3.1. Framing Tracking Error Simulation as Filtering

When discussing any statistical analysis of daily LETF returns, over a period of  $k$  days, the following length  $2k$  random vector of daily asset returns is observed:

$$\left[ R_{Index,1}, \dots, R_{Index,k}, R_{LETF,1}, \dots, R_{LETF,k} \right] \quad (7)$$

where  $R_{Index,t}$  is the daily index return on day  $t$  for the underlying index ticker  $Index$ , and  $R_{LETF,t}$  is the daily LETF return on day  $t$  for the specified LETF ticker  $LETF$ . We assume that daily LETF returns and their matching daily index returns are related through the following decomposition:

$$R_{LETF,t} = \beta_{LETF} R_{Index,t} - \text{MgmtFee}_{LETF} + \epsilon_{LETF,t} \quad (8)$$

where  $\beta_{LETF}$  is the indicated leverage multiple for  $LETF$  (constant),  $\text{MgmtFee}_{LETF}$  is the indicated rate of daily management expenses for  $LETF$  (constant), and  $\epsilon_{LETF,t}$  is the tracking error on day  $t$  for  $LETF$  (random variable). Implicitly, then, we are assuming that the following random vector captures the same information as the vector that was stated originally:

$$\left[ R_{Index,1}, \dots, R_{Index,k}, \epsilon_{LETF,1}, \dots, \epsilon_{LETF,k} \right]. \quad (9)$$

From this view of the LETF and index returns processes we develop a simulation method that is based on the following factorization of the joint density for these variables:

$$f(R_{Index,1}, \dots, R_{Index,k}, \epsilon_{LETF,1}, \dots, \epsilon_{LETF,k}) = \prod_{j=1}^k g(\epsilon_{LETF,j} | R_{Index,j}) * f(R_{Index,1}, \dots, R_{Index,k}). \quad (10)$$

The density  $g(\epsilon_{LETF,j} | R_{Index,j})$  is a filter as used in state space modeling, where (by necessity) the index returns process is designated as the observation process and the LETF returns process is designated as the state process. This runs counter to a more intuitive arrangement of designating LETF returns as functions of the state (index returns) with observation noise (tracking error).

To handle any dependence that may exist within a sequence of index return variables, we do not factorize the density  $f(R_{Index,1}, \dots, R_{Index,k})$ . Modeling all of the index return variables jointly captures their complete  $k$  day dependence structure. To account for dependence of LETF tracking error data and inter-dependence of tracking errors and index returns, in our filtering density we will additionally include  $\ell$  lagged index returns and  $\ell$  lagged tracking errors as given, where  $\ell$  is an integer valued parameter that will be selected in a later subsection. This arrangement reflects our view that LETF tracking errors have a simpler dependence structure compared to index returns (in practice  $2\ell \ll k$ ), and that LETF tracking errors are dependent on the underlying index returns. We do not consider index returns to be dependent on the respective LETF returns or tracking errors; therefore, nowhere in our approach do we attempt to model the statistical relationship of  $\epsilon_{LETF} | R_{Index}$ .

To include the additional  $2\ell$  lagged variables, the joint density factorization is revised as follows:

$$\begin{aligned} & f(R_{Index,1}, \dots, R_{Index,\ell+k}, \epsilon_{LETF,1}, \dots, \epsilon_{LETF,\ell+k}) \\ &= \prod_{j=\ell+2}^{\ell+k} g(\epsilon_{LETF,j} | R_{Index,j}, R_{Index,j-\ell}, \dots, R_{Index,j-1}, \epsilon_{LETF,j-\ell}, \dots, \epsilon_{LETF,j-1}) \\ & \quad * g(\epsilon_{LETF,1}, \dots, \epsilon_{LETF,\ell+1} | R_{Index,1}, \dots, R_{Index,\ell+1}) * f(R_{Index,1}, \dots, R_{Index,\ell+k}) \end{aligned} \quad (11)$$

There are three important changes to highlight. First, the filtering density  $g(\epsilon_{LETF,j} | R_{Index,j}, \dots)$  is now conditioned on  $\ell$  lagged index return variables and  $\ell$  lagged LETF return variables (in addition to  $R_{Index,j}$ ). Second, the joint density of index returns and LETF tracking errors (left-hand side), and unconditional joint density of index returns (final term of right-hand side), both include the additional lagged variables such that the random vector being modeled is now of length  $(\ell + k)$ . Third, we have introduced a conditional density for smoothing the  $(\ell + 1)$  tracking error variables,  $g(\epsilon_{LETF,1}, \dots, \epsilon_{LETF,\ell+1} | R_{Index,1}, \dots, R_{Index,\ell+1})$ . This additional term is necessary to initiate, or “burn-in”, the sequence of tracking error variables, providing the first set of  $\ell$  lags that are required by the first iteration of the filtering density.

Also, the decision to smooth the first  $(\ell + 1)$  variables instead of the first  $\ell$  variables and then starting

filtering at  $j = (\ell + 1)$  deserves further comment. Our reasoning is implementation related, so the dataset needs to be transformed only once. As written, the conditional densities differ only by which variables are considered as given. Otherwise both densities model the same  $(2\ell + 2)$  variables—a sequence of  $(\ell + 1)$  index returns and the matching sequence of  $(\ell + 1)$  tracking errors. Having both densities model the same variables allows for the same observation matrix being used for both of the resulting kernel density estimates, as shown in the next subsection of derivations. If only the first  $\ell$  variables were smoothed, and iterations of the filtering density began with day  $(\ell + 1)$ , the dataset would need to be transformed twice. One transform would be required to implement an estimate of the density with  $2\ell$  variables used for smoothing the initial lagged errors and another to implement an estimate of the density with  $(2\ell + 2)$  variables used for filtering the remaining errors. There is one exception, when  $\ell = 0$ , a case which results in smoothing and filtering densities that are identical and the joint density factorization reverts to the initially stated form with only two terms,

$$\left[ \prod_{j=1}^k g(\epsilon_{LETF,j} | R_{Index,j}) \right] * f(R_{Index,1}, \dots, R_{Index,k}).$$

However, to avoid the additional bias introduced by smoothing the first  $\ell$  tracking errors (conditioning on index returns forward in time relative to the tracking errors constitutes look-ahead bias), we will discard the first  $\ell$  terms when using our simulation method, making any samples generated have an effective length of  $k$ . The  $(\ell + 1)$ st term is not discarded because the associated look-ahead bias is the same as filtering (relative to this tracking error only index returns at the same time step or backward, but not forward, are given). We acknowledge that the filtering density  $g(\epsilon_{LETF,j} | R_{Index,j}, \dots)$  also constitutes the look-ahead bias. In real time we observe the daily closing index price and the daily closing LETF price (which imply daily returns) simultaneously, so the information provided by  $R_{Index,j}$  should not be available for filtering  $\epsilon_{LETF,j}$ . But we accept this bias for the increased precision of our estimate of the filtering density that it allows, thus improving our modeling of the realized volatility of LETF returns. Our method is not intended for a live setting and we do not calculate the expectation of any random variables (e.g.  $E[\epsilon_{LETF,j} | R_{Index,j}, \dots]$ ) for the purposes of predicting tracking errors or LETF returns.

### 3.2. Derivation of Kernel Density Estimates

Given a random process generating daily LETF returns  $\{R_{LETF,t}, t \in T\}$ , a random process generating the daily underlying index returns  $\{R_{Index,t}, t \in T\}$ , the target leverage multiple of the LETF ( $\beta_{LETF}$ ), the rate of daily management expenses incurred by the LETF ( $\text{MgmtFee}_{LETF}$ ), and a chosen number of lags  $\ell$  indicating the degree of dependence, we construct kernel density estimates of the smoothing density

$$g(\epsilon_{LETF,1}, \dots, \epsilon_{LETF,\ell+1} | R_{Index,1}, \dots, R_{Index,\ell+1}) \text{ and filtering density } g(\epsilon_{LETF,j} | R_{Index,j}, \dots) \text{ as follows.}$$

Note, in this subsection we use slightly different notation so that the implementation details of the kernel density estimates can be documented more clearly. The selection of  $\ell$  will be discussed in the following subsection that outlines the tracking error simulation routine.

1) Calculate the implied tracking errors from the observed data, and transform the data to allow for fitting the desired density estimates.

Assume the following relationship for decomposing daily LETF returns (as stated previously):

$$R_{LETF,t} = \beta_{LETF} R_{Index,t} - \text{MgmtFee}_{LETF} + \epsilon_{LETF,t}. \quad (12)$$

Using the decomposition, observed LETF returns ( $r_{LETF}$ ), and observed index returns ( $r_{Index}$ ), calculate the implied tracking error observations ( $e_{LETF}$ ). All of the observation vectors are length  $T$ .

Take the log transform of observed index returns  $\log(1 + r_{Index})$  and arrange them into a  $(T - \ell) \times (\ell + 1)$  observation matrix of rolling  $(\ell + 1)$  day periods. Denote this matrix as  $\mathcal{X}$ . In the same way, take the log transform of observed tracking errors  $\log(1 + e_{LETF})$  and arrange them into a  $(T - \ell) \times (\ell + 1)$  observation matrix of rolling  $(\ell + 1)$  day periods. Denote this matrix as  $\mathcal{Y}$ .

To form the final observation matrix used for fitting the kernel density estimate, combine  $\mathcal{X}$  and  $\mathcal{Y}$  in column-wise fashion. Denote this as  $\mathcal{Z}$ , a  $(T - \ell) \times (2\ell + 2)$  matrix. For notational convenience in the following steps, let  $n = T - \ell$  and  $p = 2\ell + 2$ , so that observation matrix  $\mathcal{Z}$  is  $n \times p$ .

$$\mathcal{Z} = [\mathcal{X} \quad \mathcal{Y}] \quad (13)$$



$$\begin{bmatrix} \log(1+r_{Index,1}) & \cdots & \log(1+r_{Index,\ell+1}) & \log(1+e_{LETF,1}) & \cdots & \log(1+e_{LETF,\ell+1}) \\ \log(1+r_{Index,2}) & \cdots & \log(1+r_{Index,\ell+2}) & \log(1+e_{LETF,2}) & \cdots & \log(1+e_{LETF,\ell+2}) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \log(1+r_{Index,T-\ell}) & \cdots & \log(1+r_{Index,T}) & \log(1+e_{LETF,T-\ell}) & \cdots & \log(1+e_{LETF,T}) \end{bmatrix}. \quad (14)$$

Using log returns  $(\log(1+r))$  and log tracking errors  $(\log(1+e))$  is extremely important because of the fact that log values do not suffer from asymmetry or a hard left bound at 0 (now transformed to  $-\infty$ ), making the transformed data less challenging to model.

2) Fit a kernel density estimate for  $f(\mathbf{z})$ , using a jointly independent multivariate normal kernel and observation matrix  $\mathcal{Z}$ .

$$\hat{f}(\mathbf{z}) = \frac{1}{n} \sum_{i=1}^n \phi_p(\mathbf{z} | \dot{\mathbf{z}}_{(i)}, \text{diag}(\mathbf{h}^2)) \quad (15)$$

where  $\phi_p(\mathbf{z} | \boldsymbol{\mu}, \Sigma)$  is the  $p$  dimensional multivariate normal density function evaluated at  $\mathbf{z}$  using mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\Sigma$ ,  $\dot{\mathbf{z}}_{(i)}$  is the  $i$ th row vector of observation matrix  $\mathcal{Z}$ , and  $\text{diag}(\mathbf{h}^2)$  is the diagonal matrix formed from element-wise squaring of the  $p$  dimensional vector of bandwidths. Elements indexed  $1:(\ell+1)$  correspond to bandwidths for index returns, and elements  $(\ell+2):p$  correspond to bandwidths for tracking errors.

The selection of  $\mathbf{h}$  is dependent on the LETFs (and underlying indexes) being analyzed. Our selections for the demonstration set of LETFs will be detailed later in this section.

Note that in using a diagonal matrix based on the vector of  $h_j$ s, the selected bandwidths for single dimensions, we are not ignoring the dependence structure that exists between the variables. The dependence structure will be estimated through the addition of the  $n$  multivariate kernels centered on the observation data points. This point might be easier to understand algebraically, because although the kernel in the density estimate can be decomposed as the product of univariate kernels, it is not the same as an estimate that is the product of independent univariate density estimates.

$$\frac{1}{n} \sum_{i=1}^n \phi_p(\mathbf{z} | \dot{\mathbf{z}}_{(i)}, \text{diag}(\mathbf{h}^2)) = \frac{1}{n} \sum_{i=1}^n \left[ \prod_{j=1}^p \phi(z_j | \dot{z}_{i,j}, h_j^2) \right] \neq \prod_{j=1}^p \left[ \frac{1}{n} \sum_{i=1}^n \phi(z_j | \dot{z}_{i,j}, h_j^2) \right] = \prod_{j=1}^p \hat{f}(z_j) \quad (16)$$

where  $\dot{z}_{i,j}$  is the  $j$ th element of observation vector  $\dot{\mathbf{z}}_{(i)}$ .

By capturing the dependence structure of the data through the addition of multivariate kernels that use a diagonal covariance matrix, we are achieving an effective compromise between modeling power and complexity. Selecting bandwidths for dimensions individually incorporates some information about the respective variable, yet making  $p$  separate selections separately is not difficult. The alternative choices of using a single bandwidth for all dimensions or selecting an entire covariance matrix are generally considered too simple or too difficult, respectively [20].

3) Factor and rearrange terms of the density estimate for  $f(\mathbf{z})$  to produce an estimate of the conditional density  $g(\mathbf{z}_{(\ell+2):p} | \cdots)$ .

$$g(\mathbf{z}_{(\ell+2):p} | \mathbf{z}_{1:(\ell+1)}) = f(\mathbf{z}) / f(\mathbf{z}_{1:(\ell+1)}) \quad (17)$$

where  $\mathbf{z}_{1:(\ell+1)}$  is the subvector of elements indexed  $1:(\ell+1)$  from vector  $\mathbf{z}$ , or the elements that represent index returns whose observations were sourced from the matrix  $\mathcal{X}$ , and  $\mathbf{z}_{(\ell+2):p}$  is the subvector of elements indexed  $(\ell+2):p$  from vector  $\mathbf{z}$ , or the elements that represent tracking errors whose observations were sourced from the matrix  $\mathcal{Y}$ . Equation (17) could be restated as  $g(\mathbf{y} | \mathbf{x}) = f(\mathbf{x}, \mathbf{y}) / f(\mathbf{x})$ ; however, the following notation will be used for the sake of brevity and consistency with Step 4.

$$g(\mathbf{z}_y | \mathbf{z}_x) = f(\mathbf{z}) / f(\mathbf{z}_x) \quad (18)$$

In this way, when  $x$  and  $y$  are written as subscripts, let  $\mathbf{z}_x = \mathbf{z}_{1:(\ell+1)}$  and  $\mathbf{z}_y = \mathbf{z}_{(\ell+2):p}$ , and by extension  $\mathbf{h}_x^2 = \mathbf{h}_{1:(\ell+1)}^2$  and  $\mathbf{h}_y^2 = \mathbf{h}_{(\ell+2):p}^2$ .

Continue by inserting the appropriate density estimates.

$$\hat{g}(\mathbf{z}_y | \mathbf{z}_x) = \hat{f}(\mathbf{z}) / \hat{f}(\mathbf{z}_x) \quad (19)$$

$$= \left[ \frac{1}{n} \sum_{i=1}^n \phi_p(\mathbf{z} | \dot{\mathbf{z}}_{(i)}, \text{diag}(\mathbf{h}^2)) \right] / \left[ \frac{1}{n} \sum_{i=1}^n \phi_{\ell+1}(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) \right] \quad (20)$$

Because of our choice of a jointly independent multivariate normal kernel,  $\phi_p(\mathbf{z} | \dots)$  can be factored as follows:

$$\phi_p(\mathbf{z} | \dot{\mathbf{z}}_{(i)}, \text{diag}(\mathbf{h}^2)) = \phi_{\ell+1}(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) * \phi_{\ell+1}(\mathbf{z}_y | \dot{\mathbf{z}}_{(i),y}, \text{diag}(\mathbf{h}_y^2)). \quad (21)$$

Insert this factorization and rearrange terms to produce the final density estimate.

$$\hat{g}(\mathbf{z}_y | \mathbf{z}_x) = \left[ \frac{1}{n} \sum_{i=1}^n \phi_{\ell+1}(\mathbf{z}_y | \dot{\mathbf{z}}_{(i),y}, \text{diag}(\mathbf{h}_y^2)) * \phi_{\ell+1}(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) \right] / \left[ \frac{1}{n} \sum_{i=1}^n \phi_{\ell+1}(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) \right] \quad (22)$$

$$\hat{g}(\mathbf{z}_y | \mathbf{z}_x) = \sum_{i=1}^n \left[ \phi_{\ell+1}(\mathbf{z}_y | \dot{\mathbf{z}}_{(i),y}, \text{diag}(\mathbf{h}_y^2)) * \alpha_i(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) \right] \quad (23)$$

$$\alpha_i(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) = \phi_{\ell+1}(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) / \left[ \frac{1}{n} \sum_{i=1}^n \phi_{\ell+1}(\mathbf{z}_x | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) \right]. \quad (24)$$

$\hat{g}(\mathbf{z}_y | \mathbf{z}_x)$  is the estimate of the smoothing density that will be used to simulate the first  $(\ell+1)$  tracking errors of a new sample.

4) Factor and rearrange terms of the density estimate for  $f(\mathbf{z})$  to produce an estimate of the conditional density  $g(\mathbf{z}_p | \dots)$ .

$$g(\mathbf{z}_p | \mathbf{z}_{-p}) = f(\mathbf{z}) / f(\mathbf{z}_{-p}) \quad (25)$$

where  $\mathbf{z}_{-p}$  is the vector  $\mathbf{z}$  without the last element. Equation (25) could be restated as

$g(y_{\ell+1} | \mathbf{x}, \mathbf{y}_{-(\ell+1)}) = f(\mathbf{x}, \mathbf{y}) / f(\mathbf{x}, \mathbf{y}_{-(\ell+1)})$ ; however, the original notation will be maintained for the sake of brevity.

Continue by inserting the appropriate density estimates.

$$\hat{g}(\mathbf{z}_p | \mathbf{z}_{-p}) = \hat{f}(\mathbf{z}) / \hat{f}(\mathbf{z}_{-p}) \quad (26)$$

$$= \left[ \frac{1}{n} \sum_{i=1}^n \phi_p(\mathbf{z} | \dot{\mathbf{z}}_{(i)}, \text{diag}(\mathbf{h}^2)) \right] / \left[ \frac{1}{n} \sum_{i=1}^n \phi_{p-1}(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) \right] \quad (27)$$

As in Step 3, factor  $\phi_p(\mathbf{z} | \dots)$ , but adjusting for the desired elements of  $\mathbf{z}$ .

$$\phi_p(\mathbf{z} | \dot{\mathbf{z}}_{(i)}, \text{diag}(\mathbf{h}^2)) = \phi(\mathbf{z}_p | \dot{\mathbf{z}}_{i,p}, h_p^2) * \phi_{p-1}(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) \quad (28)$$

Insert this factorization and rearrange terms to produce the final density estimate.

$$\hat{g}(\mathbf{z}_p | \mathbf{z}_{-p}) = \left[ \frac{1}{n} \sum_{i=1}^n \phi(\mathbf{z}_p | \dot{\mathbf{z}}_{i,p}, h_p^2) * \phi_{p-1}(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) \right] / \left[ \frac{1}{n} \sum_{i=1}^n \phi_{p-1}(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) \right] \quad (29)$$

$$\hat{g}(\mathbf{z}_p | \mathbf{z}_{-p}) = \sum_{i=1}^n \left[ \phi(\mathbf{z}_p | \dot{\mathbf{z}}_{i,p}, h_p^2) * \alpha_i(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) \right] \quad (30)$$

$$\alpha_i(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) = \phi_{p-1}(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) / \left[ \frac{1}{n} \sum_{i=1}^n \phi_{p-1}(\mathbf{z}_{-p} | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) \right] \quad (31)$$

$\hat{g}(\mathbf{z}_p | \mathbf{z}_{-p})$  is the estimate of the filtering density that will be used to simulate tracking errors  $(\ell+2)$  through  $k$  of a new sample.

### 3.3. Simulation Routine

With the final density estimates ready, LETF tracking errors can be simulated using the following routine. Given:

length  $(\ell + k)$  input vector of log index returns  $\mathbf{u}$ , length  $p$  bandwidth vector  $\mathbf{h}$ , number of daily lags  $\ell$ , and length  $p$  observation vectors  $\dot{\mathbf{z}}_{(1)}, \dots, \dot{\mathbf{z}}_{(n)}$

- 1) Sample an index value  $i \in 1, \dots, n$  from a multinomial distribution with weight vector  $\boldsymbol{\alpha}$ , where  $\alpha_i = \phi_{\ell+1}(\mathbf{u}_{1:(\ell+1)} | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) / \left[ \sum_{i=1}^n \phi_{\ell+1}(\mathbf{u}_{1:(\ell+1)} | \dot{\mathbf{z}}_{(i),x}, \text{diag}(\mathbf{h}_x^2)) \right]$ .
2. Use index value  $i$  from step 1 to select the  $i$ th observation vector  $\dot{\mathbf{z}}_{(i)}$  and sample  $\mathbf{v}_{1:(\ell+1)}$ , an  $(\ell + 1)$  length vector from the multivariate normal distribution with density  $\phi_{\ell+1}(\cdot | \dot{\mathbf{z}}_{(i),y}, \text{diag}(\mathbf{h}_y^2))$ .

Perform steps 3 and 4 for  $j = (\ell + 2), \dots, (\ell + k)$

- 3) Sample an index value  $i \in 1, \dots, n$  from a multinomial distribution with weight vector  $\boldsymbol{\alpha}$ , where  $\alpha_i = \phi_{p-1}(\left[ \mathbf{u}_{(j-\ell):j}, \mathbf{v}_{(j-\ell):(j-1)} \right] | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) / \left[ \sum_{i=1}^n \phi_{p-1}(\left[ \mathbf{u}_{(j-\ell):j}, \mathbf{v}_{(j-\ell):(j-1)} \right] | \dot{\mathbf{z}}_{(i),-p}, \text{diag}(\mathbf{h}_{-p}^2)) \right]$ , and  $\left[ \mathbf{u}_{(j-\ell):j}, \mathbf{v}_{(j-\ell):(j-1)} \right]$  is the length  $(p-1)$  vector formed by concatenating the indicated elements of  $\mathbf{u}$  and  $\mathbf{v}$ .
- 4) Use index value  $i$  from step 3 to select the  $i$ th observation vector  $\dot{\mathbf{z}}_{(i)}$  and sample  $\mathbf{v}_j$  from the multivariate normal distribution with density  $\phi(\cdot | \dot{\mathbf{z}}_{(i),p}, h_p^2)$ .
- 5) Combine the simulated tracking errors  $\mathbf{v}_{(\ell+1):(\ell+k)}$  with the input index returns  $\mathbf{u}_{(\ell+1):(\ell+k)}$  according to our LETF returns decomposition.

$$w_t = \beta_{LETF} * (\exp(u_t) - 1) - \text{MgmtFee}_{LETF} + (\exp(v_t) - 1) \quad (32)$$

where  $\mathbf{w}$  is the final, length  $k$  output vector of simulated LETF returns. Elements indexed  $1:\ell$  in vectors  $\mathbf{u}$ ,  $\mathbf{v}$ , and  $\mathbf{w}$  are discarded.

When simulating data for the intended application, the sampled index returns to be used as inputs are first arranged as a matrix of rolling periods of  $(\ell + k)$  days (and transformed to log returns). Regular choices of  $k$  are 21 or 252. The simulation routine can be executed in parallel because there is no dependence on the input data across rows. Each row of the matrix is input as vector  $\mathbf{u}$  separately and the density sampling steps can execute on each row simultaneously. Step 5 can obviously execute in parallel with basic matrix algebra. The simulation routine is also put to use for estimating the lag parameter  $\ell$ , but the input data is not arranged into rolling periods initially. With lag estimation this step occurs after sampling using the simulated data in order to avoid simulating multiple tracking errors based on the same index returns.

### 3.4. Lag and Bandwidth Selection for Simulation of LETF Tracking Errors

Estimating the lag parameter  $\ell$  proceeds as a brute force sweep of possible selections 0 to  $N$  (for some reasonable  $N$ ), using the simulation routine and a Kolmogorov-Smirnov test for evaluation of fit. For each LETF and index combination and choice of  $\ell$  to be considered,  $K$  iterations of tracking error simulation are performed (for some large  $K$ ). The entire observation vector of daily index returns (log transformed) for the days which LETF returns are available is supplied as the input vector  $\mathbf{u}$ . The routine is executed without the use of rolling periods, but this transformation is made after Step 5 has completed, such that the simulated LETF returns are arranged into a matrix of rolling periods of  $k$  returns. The compound return is calculated for each row producing a final simulated sample of  $k$  day LETF returns. These two steps are repeated with the actual LETF returns to produce the reference sample of  $k$  day LETF returns, and then the samples are compared using a two-tailed Kolmogorov-Smirnov test. Final lag selection is based on the resulting  $p$ -values collected from  $K$  iterations of the simulation routine and Kolmogorov-Smirnov test.

The K-S test is comparing a sample of  $k$  day LETF returns derived from observed index returns and simulated LETF tracking errors against a sample of  $k$  day LETF returns derived from observed index returns and observed LETF tracking errors (*i.e.* the observed LETF returns). Each iteration uses only the index returns observed on days when the paired LETF was trading to ensure that the two samples are derived from the same sequence of data. The null hypothesis in a K-S test states that the two samples being compared are drawn from the same distribution [21]. Our goal is to simulate tracking errors in a way that the K-S test of the LETF returns samples cannot reject the null hypothesis at sufficiently high confidence levels. Based on the results of the K-S tests, our goal translates to selecting the value of  $\ell$  that produces greater  $p$ -values compared to other selections. This

generic mandate could be interpreted in many different ways, and we settled on selecting the value of  $\ell$  with the greatest proportion of  $p$ -values that were  $\geq 0.01$ . For some LETFs there were multiple possible  $\ell$  selections where all resulting  $p$ -values were  $\geq 0.01$ . In these cases ties were broken by selecting the least value of  $\ell$  to avoid any possible overfitting. A  $p$ -value that is  $\geq 0.01$  indicates that iteration of simulation produced a sample that could not be rejected at a confidence level  $\geq 0.99$ , the threshold of confidence we are in effect selecting to delineate as sufficiently high.

In **Table 1**, we present our final lag selections for the demonstration set of LETFs. Column headers prefixed with “21D” or “252D” refer to results produced when using  $k = 21$  day or  $k = 252$  day periods, respectively, and the notation  $P[p \geq 0.01]$  refers to the proportion of  $p$ -values that were  $\geq 0.01$ . We used 10,000 iterations for testing each LETF. We feel that our selection criteria resulted in satisfactory lag parameter selections for all LETFs when using 21 day periods (except for FINZ), but the results are mixed when using 252 day periods, with 9 of 20 LETFs (FINZ, DUST, TECL, TECS, RUSL, RUSS, DZK, EDC, EDZ) failing to offer values of  $P[p \geq 0.01]$  greater than 0.9. There were many lag selections for each LETF that performed well when using 21 day periods, so we simplified the estimation task and present only the lag selection that provided the best results with 252 day periods (the values shown in columns with a “21D” prefix may not actually be the best). The minimum and median  $p$ -values were not incorporated into any decision making, but are included here for reference.

Our selections for the kernel bandwidth vector  $\mathbf{h}$  are based on the multivariate “Scott Rule” [20], defined as:

**Table 1.** Final lag selections for all demonstration LETFs.

Index	Leverage	LETF	Lags	21D	21D	21D	252D	252D	252D
				P [ $p \geq 0.01$ ]	Minimum	Median	P [ $p \geq 0.01$ ]	Minimum	Median
DJUSFN	+3	FINU	3	1.0000	0.7125	0.9963	1.0000	0.0354	0.6315
DJUSFN	−3	FINZ	3	0.6034	0.0000	0.0242	0.0511	0.0000	0.0000
GDM	+3	NUGT	5	1.0000	0.1592	0.4248	0.9986	0.0053	0.0644
GDM	−3	DUST	3	1.0000	0.0466	0.3639	0.5112	0.0000	0.0107
INDU	+3	UDOW	3	1.0000	0.9962	1.0000	1.0000	0.5142	1.0000
INDU	−3	SDOW	3	1.0000	0.9993	1.0000	1.0000	0.4442	0.9994
IXT	+3	TECL	4	0.9996	0.0059	0.5625	0.5942	0.0000	0.0450
IXT	−3	TECS	5	0.9953	0.0006	0.3224	0.5057	0.0000	0.0107
MVRSX	+3	RUSL	3	0.9997	0.0086	0.7218	0.1339	0.0000	0.0000
MVRSX	−3	RUSS	4	1.0000	0.0183	0.7212	0.5420	0.0000	0.0180
MXEA	+3	DZK	3	0.9999	0.0067	0.6241	0.0268	0.0000	0.0000
MXEA	−3	DPK	4	1.0000	0.2803	0.9546	0.9429	0.0000	0.1643
MXEF	+3	EDC	3	1.0000	0.0411	0.3948	0.0455	0.0000	0.0000
MXEF	−3	EDZ	4	1.0000	0.2243	0.9398	0.6698	0.0000	0.0403
NDX	+3	TQQQ	3	1.0000	0.9997	1.0000	1.0000	0.6649	1.0000
NDX	−3	SQQQ	3	1.0000	0.9999	1.0000	1.0000	0.5509	0.9983
RTY	+3	TNA	3	1.0000	0.9582	1.0000	1.0000	0.0911	0.9682
RTY	−3	TZA	4	1.0000	0.9986	1.0000	1.0000	0.0121	0.8987
SPX	+3	SPXL	5	1.0000	0.9912	1.0000	1.0000	0.0605	1.0000
SPX	−3	SPXS	5	1.0000	0.9912	1.0000	1.0000	0.1100	1.0000

$$\hat{h}_j = \hat{\sigma}_j * n^{-1/(p+4)} \quad (33)$$

where  $\hat{h}_j$  is the bandwidth for  $j$ th dimension of density estimate,  $\hat{\sigma}_j$  is the estimated standard deviation of  $j$ th dimension,  $n$  is the number of  $p$  dimensional observations used for fitting density estimate, and  $p$  is the number of dimensions of the sample space.

We use  $s^2$ , the traditional unbiased estimator of  $\sigma^2$ , and scale down our final selections of  $\hat{h}_j$ . After extensive testing of our tracking error simulation routine with the demonstration set of 20 LETFs, we concluded that the elements of  $\hat{\mathbf{h}}$  corresponding to index returns (elements  $1:(\ell+1)$ ) needed to be scaled down by a factor of 100, and the elements of  $\hat{\mathbf{h}}$  corresponding to LETF tracking errors (elements  $(\ell+2):p$ ) needed to be scaled down by a factor of 100,000. We provide visual justification for this heuristic in **Figure 4**. For the final bandwidth selections, we have:

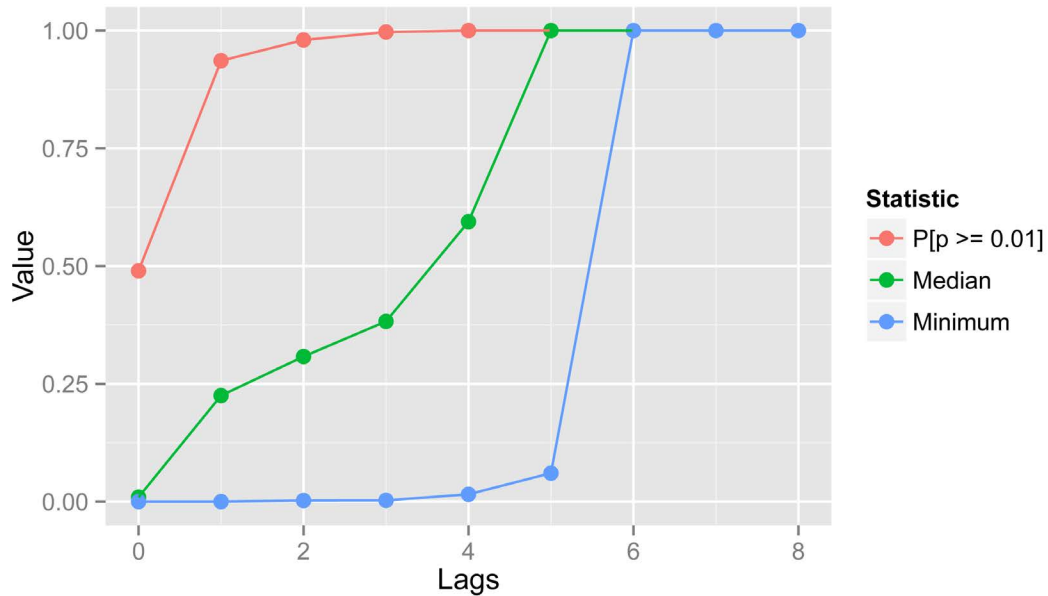
$$\hat{h}_j = s_j * n^{-1/(p+4)} * 0.01, \text{ for } j = 1, \dots, (\ell+1) \quad (34)$$

$$\hat{h}_j = s_j * n^{-1/(p+4)} * 0.000001, \text{ for } j = (\ell+2), \dots, p \quad (35)$$

$$s_j = \left[ \frac{1}{n-1} \sum_{i=1}^n (z_{i,j} - \bar{z}_j)^2 \right]^{1/2} \quad (36)$$

$$\bar{z}_j = \frac{1}{n} \sum_{i=1}^n z_{i,j}. \quad (37)$$

**Figure 4** shows the lag testing results from 10,000 iterations of LETF Ticker SPXL using 252 day periods for each lag parameter value 0 through 8. In all simulations, the bandwidths were scaled down as indicated above, producing in our opinion a reasonable increase in K-S  $p$ -values (using any of the statistics shown) as the lag value increases from 0 to 8. These results are somewhat representative of the results experienced when testing the other LETFs. Alternatively, when using larger factors to scale the bandwidths, the simulation routine would fail on some iterations because the estimates for the smoothing and filtering densities did not have sufficient support. We experienced this type of result when testing FINZ, TECS, RUSL, and RUSS with the factors indicated above, forcing us to create an exception for these LETFs where we decrease the factor used to scale the tracking error portion of the bandwidth vector to 10,000. When using smaller factors to scale the bandwidths, significant increases in  $p$ -values were generally not observed at any lag, or not observed until 8 or more lags were included. In the latter cases, these results appeared to us as overfitting.



**Figure 4.** Lag testing results summary for LETF ticker SPXL with 252 day periods.

## 4. Constrained Simulation Using Nonparametric Density Estimation

The nonparametric method for constrained simulation we propose in this section was devised as part of the larger body of work on LETF volatility documented throughout this paper, but the method is truly general in nature. It could be helpful in any setting where the practitioner would like to simulate multivariate data with a constrained sample sum (or average). For this reason the second and third subsections are presented in the context of using generic multivariate data, without references to financial data or financial constructs.

### 4.1. Extending the LETF Returns Model with an Index Return Constraint

In order to constrain the compound underlying index return for the period, the joint density of our LETF returns model from Section 3 must be revised as follows:

$$\begin{aligned} f(R_{Index,1}, \dots, R_{Index,\ell+k-1}, \epsilon_{LETF,1}, \dots, \epsilon_{LETF,\ell+k} | R_0) = \\ \prod_{j=\ell+2}^{\ell+k} g(\epsilon_{LETF,j} | R_{Index,j}, R_{Index,j-\ell}, \dots, R_{Index,j-1}, \epsilon_{LETF,j-\ell}, \dots, \epsilon_{LETF,j-1})^* \\ g(\epsilon_{LETF,1}, \dots, \epsilon_{LETF,\ell+1} | R_{Index,1}, \dots, R_{Index,\ell+1})^* f(R_{Index,1}, \dots, R_{Index,\ell+k-1} | R_0) \end{aligned} \quad (38)$$

where  $R_0 = \prod_{j=\ell+1}^{\ell+k} (1 + R_{Index,j}) - 1$ , the total  $k$  day compounded return of the underlying index for days  $(\ell+1)$  through  $(\ell+k)$ , and the density function  $f(R_{Index,1}, \dots, R_{Index,\ell+k-1} | R_0)$  is the joint density of a sequence of  $\ell$  unconstrained index return variables (the lags) and a sequence of  $(k-1)$  index return variables constrained to compound to  $R_0$ . This form of the full joint density for index returns and LETF tracking errors (left-hand side) can be used to simulate sequences of LETF returns constrained such that the sampled sequence of index returns compounds to the given total period return  $R_0$ .

It needs to be emphasized that this modification is implemented using the log form of index returns as was the case with all estimators and simulation work featured in Section 3. When index return variables are modeled as  $X = \log(1 + R_{Index})$ , the density being estimated becomes  $f(X_1, \dots, X_{\ell+k-1} | X_0 = \sum_{j=\ell+1}^{\ell+k-1} X_j)$ , the joint density of a sequence of  $\ell$  unconstrained variables (the lags) and a sequence of  $(k-1)$  variables constrained to sum to  $X_0$ . The estimate for this form of the constrained joint density is much easier to derive and more generally useful in applications outside of finance.

### 4.2. Derivation of Kernel Density Estimate

Given any  $p$  dimensional random vector  $\mathbf{X}$  with  $\ell \in [0, \dots, (p-2)]$  unconstrained dimensions (lags) and  $n$  observations of  $\mathbf{X}$  arranged as an  $(n \times p)$  observation matrix  $\mathcal{X}$ , our nonparametric density estimate is constructed as follows:

1) Fit a kernel density estimate for  $f(\mathbf{x})$  using a jointly independent multivariate normal kernel and observation matrix  $\mathcal{X}$ .

$$\hat{f}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^n \phi_p(\mathbf{x} | \dot{\mathbf{x}}_{(i)}, \text{diag}(\mathbf{h}^2)) \quad (39)$$

where  $\phi_p(\mathbf{x} | \boldsymbol{\mu}, \Sigma)$  is the  $p$  dimensional multivariate normal density function evaluated at  $\mathbf{x}$  using mean vector  $\boldsymbol{\mu}$  and covariance matrix  $\Sigma$ ,  $\dot{\mathbf{x}}_{(i)}$  is the  $i$ th row vector of observation matrix  $\mathcal{X}$ , and  $\text{diag}(\mathbf{h}^2)$  is the diagonal matrix formed from element-wise squaring of the  $p$  dimensional vector of bandwidths.

Note that the selection of  $\mathbf{h}$  is ultimately application dependent, so we cannot provide advice for any kind of general case other than recommending what has already been published about using the multivariate normal density as a kernel [20]. The selections for our applications will be detailed later in a later subsection.

2) Perform a change of variables and derive the corresponding kernel density estimate.

Let  $Y_1 = X_1$ ,  $Y_2 = X_2$ ,  $Y_3 = X_3$ ,  $\dots$ ,  $Y_p = \sum_{j=\ell+1}^p X_j$ . Or using matrix notation, define random vector  $\mathbf{Y}$  as follows:

$$\mathbf{Y} = \mathbf{T}\mathbf{X} \quad (40)$$



$$\mathcal{T} = \begin{bmatrix} 1 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \ddots & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 & 1 & \cdots & 1 \end{bmatrix}, \text{ a } (p \times p) \text{ matrix} \quad (41)$$

where the first  $\ell$  elements of the bottom row of  $\mathcal{T}$  are 0 to allow for the unconstrained dimensions of  $\mathbf{X}$ .

To derive the kernel density estimate for the transformed variables, invoke the following theorem for transforming multivariate normal random vectors [22]:

$$\text{If } \mathbf{X} \sim \phi(\mathbf{x} | \boldsymbol{\mu}, \Sigma) \text{ and } \mathbf{Y} = \mathcal{A}\mathbf{X} + \mathbf{c}, \text{ then } \mathbf{Y} \sim \phi(\mathbf{y} | \mathcal{A}\boldsymbol{\mu} + \mathbf{c}, \mathcal{A}\Sigma\mathcal{A}') \quad (42)$$

where  $\mathcal{A}$  is a constant matrix,  $\mathcal{A}'$  is the transpose of that matrix, and  $\mathbf{c}$  is a constant vector. Apply the theorem to the kernel from the result of Step 1 to produce the estimate of the joint density for  $\mathbf{Y}$ .

$$\hat{f}(\mathbf{y}) = \frac{1}{n} \sum_{i=1}^n \phi_p(\mathbf{y} | \mathcal{T} \dot{\mathbf{x}}_{(i)}, \mathcal{T} \text{diag}(\mathbf{h}^2) \mathcal{T}') \quad (43)$$

3) Factor the kernel within  $\hat{g}(\mathbf{y})$  into the product of an unconditional univariate kernel for  $y_p$  and a multivariate kernel for  $y_1, y_2, \dots, y_{p-1}$  conditional on  $y_p$ .

$$\phi_p(\mathbf{y} | \dots) = \phi(y_p | \dots) \phi_{p-1}(\mathbf{y}_{-p} | y_p, \dots) \quad (44)$$

where  $\mathbf{y}_{-p}$  is the vector  $\mathbf{y}$  without the  $p$ th element.

For the kernel factor corresponding to the univariate density for  $Y_1$ , re-apply the theorem cited in Step 2 to construct the following univariate density kernel:

$$\phi(y_p | \dots) = \phi\left(y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2\right). \quad (45)$$

To construct  $\phi_{p-1}(\mathbf{y}_{-p} | y_p, \dots)$  invoke another theorem for transforming multivariate normal random vectors [22]:

$$\text{If } \mathbf{X} \sim \phi(\mathbf{x} | \boldsymbol{\mu}, \Sigma), \text{ then } \mathbf{X}_{-1} | X_1 \sim \phi_{p-1}\left(\mathbf{x}_{-1} | \boldsymbol{\mu}_{-1} + \Sigma_{2,1}(\sigma_1^2)^{-1}(x_1 - \mu_1), \Sigma_{2,2} - \Sigma_{2,1}(\sigma_1^2)^{-1}\Sigma_{1,2}\right) \quad (46)$$

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \Sigma_{1,2} \\ \Sigma_{2,1} & \Sigma_{2,2} \end{bmatrix} \quad (47)$$

where  $\sigma_1^2$ ,  $\Sigma_{1,2}$ ,  $\Sigma_{2,1}$ , and  $\Sigma_{2,2}$  taken together constitute a partition of  $\Sigma$ . Apply this theorem to the conditional kernel.

$$\phi_{p-1}(\mathbf{y}_{-p} | y_p, \dots) = \phi_{p-1}(\mathbf{y}_{-p} | \tilde{\mathbf{x}}_{(i),-p}, \tilde{\mathcal{H}}) \quad (48)$$

$$\tilde{\mathbf{x}}_{(i),-p} = \dot{\mathbf{x}}_{(i),-p} + \mathbf{h}_{-p}^2 \left( \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right)^{-1} \left( y_p - \sum_{j=\ell+1}^p \dot{x}_{i,j} \right) \quad (49)$$

$$\tilde{\mathcal{H}} = \text{diag}(\mathbf{h}_{-p}^2) - \mathbf{h}_{-p}^2 \left( \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right)^{-1} \mathbf{h}_{-p}' \quad (50)$$

where  $\dot{\mathbf{x}}_{(i),-p}$  is the  $i$ th observation vector without the  $p$ th element, and  $\mathbf{h}_{-p}'$  is the bandwidth vector  $\mathbf{h}$ , without the  $p$ th element, squared element-wise, and transposed.

4) Factor and rearrange terms of the density estimate for  $f(\mathbf{y})$  to produce an estimate for the conditional density  $g(\mathbf{y}_{-p} | y_p)$ . Start by factoring the density for  $f(\mathbf{y})$  and inserting the appropriate density estimates.

$$g(\mathbf{y}_{-p} | y_p) = f(\mathbf{y}) / f(y_p) \quad (51)$$

$$\hat{g}(\mathbf{y}_{-p} | y_p) = \hat{f}(\mathbf{y}) / \hat{f}(y_p) \quad (52)$$

$$\hat{g}(\mathbf{y}_{-p} | y_p) = \frac{1}{n} \sum_{i=1}^n \phi_p(\mathbf{y} | \mathcal{T}\mathbf{x}_{(i)}, \mathcal{T}\text{diag}(\mathbf{h}^2) \mathcal{T}') / \left[ \frac{1}{n} \sum_{i=1}^n \phi \left( y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) \right]. \quad (53)$$

Continue by inserting the factors of  $\phi_p(\mathbf{y} | \dots)$  that were derived in Step 3 and rearranging terms to produce the final density estimate.

$$\hat{g}(\mathbf{y}_{-p} | y_p) = \frac{1}{n} \sum_{i=1}^n \left[ \phi \left( y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) * \phi_{p-1}(\mathbf{y}_{-p} | \tilde{\mathbf{x}}_{(i),-p}, \tilde{\mathcal{H}}) \right] / \left[ \frac{1}{n} \sum_{i=1}^n \phi \left( y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) \right] \quad (54)$$

$$\hat{g}(\mathbf{y}_{-p} | y_p) = \sum_{i=1}^n \left[ \phi_{p-1}(\mathbf{y}_{-p} | \tilde{\mathbf{x}}_{(i),-p}, \tilde{\mathcal{H}}) * \alpha_i \phi \left( y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) \right] \quad (55)$$

$$\alpha_i \left( y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) = \phi \left( y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) / \left[ \sum_{i=1}^n \phi \left( y_p | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) \right]. \quad (56)$$

Because there was no effective change in the  $X_1, \dots, X_{p-1}$  to  $Y_1, \dots, Y_{p-1}$  transformations, define  $x_0 = y_p = \sum_{j=\ell+1}^p x_j$  to denote the given value for the sum of the random vector and the density estimate can be written entirely in terms of  $\mathbf{X}$ . The final form of the density estimate:

$$\hat{g}(\mathbf{x}_{-p} | x_0) = \sum_{i=1}^n \left[ \phi_{p-1}(\mathbf{x}_{-p} | \tilde{\mathbf{x}}_{(i),-p}, \tilde{\mathcal{H}}) * \alpha_i \left( x_0 | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) \right] \quad (57)$$

$$\alpha_i \left( x_0 | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) = \phi \left( x_0 | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) / \left[ \sum_{i=1}^n \phi \left( x_0 | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) \right] \quad (58)$$

### 4.3. Simulation Routine

With the final density estimates ready, constrained data can be simulated using the following routine.

Given: length  $p$  bandwidth vector  $\mathbf{h}$ , vector sum  $\sum_{j=\ell+1}^p x_j$  constraint  $x_0$ , and length  $p$  observation vectors  $\dot{\mathbf{x}}_{(1)}, \dots, \dot{\mathbf{x}}_{(n)}$

1) Sample an index value  $i \in 1, \dots, n$  from a multinomial distribution with weight vector  $\alpha$ , where  $\alpha_i = \phi \left( x_0 | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) / \left[ \sum_{i=1}^n \phi \left( x_0 | \sum_{j=\ell+1}^p \dot{x}_{i,j}, \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right) \right]$ .

2) Use index value  $i$  from step 1 to select the  $i$ th observation  $\dot{\mathbf{x}}_{(i)}$  and sample  $\mathbf{x}_{-p}$ , a  $(p-1)$  dimensional vector, from the multivariate normal distribution with density  $\phi_{p-1}(\cdot | \tilde{\mathbf{x}}_{(i),-p}, \tilde{\mathcal{H}})$ , where

$$\tilde{\mathbf{x}}_{(i),-p} = \dot{\mathbf{x}}_{(i),-p} + \mathbf{h}_{-p}^2 \left( \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right)^{-1} \left( x_0 - \sum_{j=\ell+1}^p \dot{x}_{i,j} \right), \text{ and } \tilde{\mathcal{H}} = \text{diag}(\mathbf{h}_{-p}^2) - \mathbf{h}_{-p}^2 \left( \sum_{j=\ell+1}^p \mathbf{h}_j^2 \right)^{-1} \mathbf{h}_{-p}^2.$$

3) Use elements  $(\ell+1)$  through  $(p-1)$  of the sampled  $(p-1)$  dimensional vector  $\mathbf{x}_{-p}$  from Step 2 to arrive at the  $p$ th (implied) sample value  $x_p = x_0 - \sum_{j=\ell+1}^{p-1} x_j$ .

Because Steps 1 and 3 are computationally trivial, the computational complexity of the simulation routine is no worse than that of the chosen implementation for the sampled multivariate normal distribution. Our testing and development was performed using the `mvrnorm()` implementation available in the R package MASS. Package version 7.3 - 27, R version 3.0.1.

**Table 2** is presented to better illustrate what real time performance a user can expect with the routine (based on our choice of `mvnrm()`). The routine was executed 100 times each using various sizes of the observation matrix ( $n \times p$ ), and for the sake of completeness included various values for number of iterations  $k$ . For all simulations the returns data for the S&P 500 index were used with a constrained return of 0 and our final bandwidth selection (described in the next subsection). As the iterations of the routine can be executed in parallel, theoretically, the number of iterations  $k$  should have no impact on the final runtime. The tests were performed on a Dell server with 2 Intel Xeon 2.5 GHz processors (6 cores each, 64 GB memory total) running a virtualized instance of Red Hat Enterprise Linux Server 6.5 (Kernel 2.6, 64 bit).

As expected there are significant increases in execution time at the largest selections for  $n$  and  $p$ , while  $k$  does not have much of an impact. With an average execution time near 3 minutes when using a  $(100000 \times 1000)$

**Table 2.** Execution Times from 100 iterations per parameter set (seconds).

$n$	$p$	$k$	mean	min	max
1000	10	1000	0.01	0.01	0.01
1000	10	10,000	0.05	0.05	0.10
1000	10	100,000	0.70	0.67	0.84
1000	100	1000	0.05	0.05	0.06
1000	100	10,000	0.51	0.42	0.57
1000	100	100,000	3.79	3.63	4.01
1000	1000	1000	4.50	4.17	4.67
1000	1000	10,000	13.95	13.72	14.29
1000	1000	100,000	108.81	108.24	110.41
10,000	10	1000	0.05	0.04	0.05
10,000	10	10,000	0.10	0.09	0.16
10,000	10	100,000	0.84	0.68	0.96
10,000	100	1000	0.24	0.23	0.31
10,000	100	10,000	0.71	0.68	0.77
10,000	100	100,000	3.99	3.93	4.15
10,000	1000	1000	11.24	11.07	12.23
10,000	1000	10,000	20.67	20.35	21.76
10,000	1000	100,000	115.92	114.91	117.25
100,000	10	1000	0.46	0.40	0.53
100,000	10	10,000	0.51	0.48	0.58
100,000	10	100,000	1.16	1.05	1.27
100,000	100	1000	2.41	2.33	2.63
100,000	100	10,000	2.77	2.73	2.91
100,000	100	100,000	6.14	6.01	6.28
100,000	1000	1000	80.94	79.40	91.49
100,000	1000	10,000	89.85	88.91	102.12
100,000	1000	100,000	185.86	184.00	197.93

observation matrix we hope that interested readers will not consider runtime as a drawback against using our routine in practice.

#### 4.4. Bandwidth Selection for Constrained Simulation of Index Returns

Our selection here is similar to the final rule presented in Section 3. Again we begin with the multivariate “Scott Rule” [20], defined as it was previously.

In the case of simulating only index returns, we simplify the initial rule because the  $p$  dimensional row vectors of the observation matrix used to fit the density estimate are rolling  $p$  day intervals of the same series of daily data. Therefore, a single value from the average of the  $\hat{\sigma}_j$  terms will be used because individually they will be nearly identical. Also, we use  $s^2$ , the traditional unbiased estimator of  $\sigma^2$ .

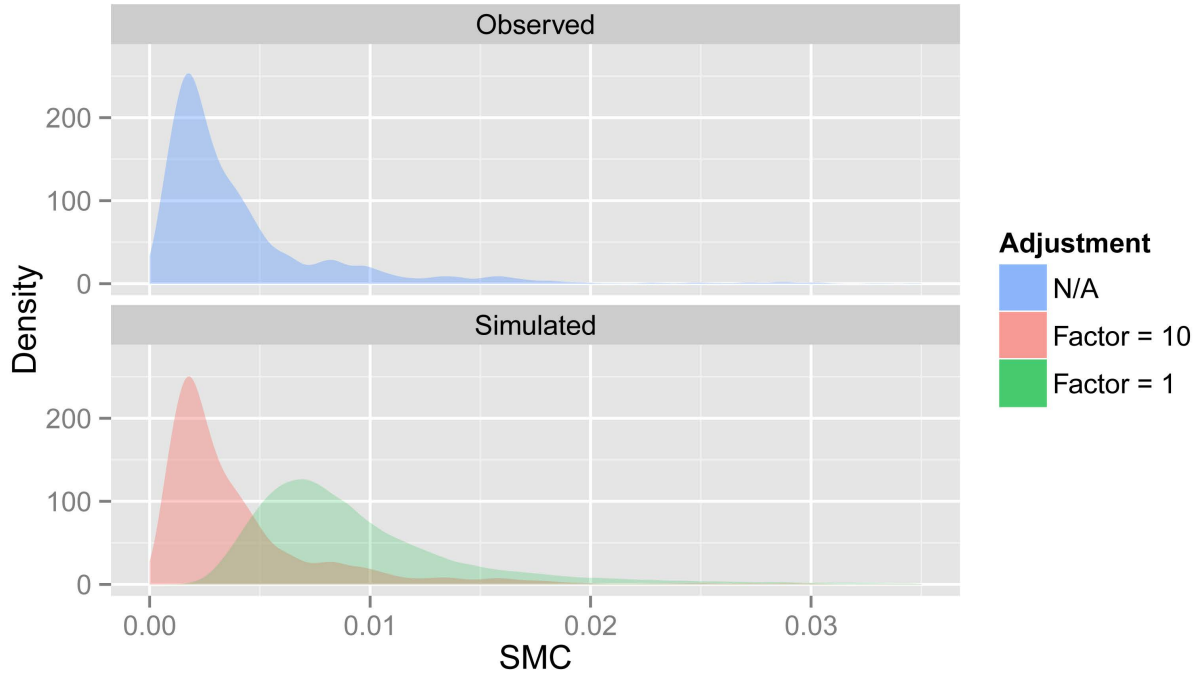
$$\hat{\mathbf{h}} = \left[ \left( \frac{1}{p} \sum_{j=1}^p s_j \right) * n^{-1/(p+4)} \right] * \mathbf{1}_p \quad (59)$$

$$s_j = \left[ \frac{1}{n-1} \sum_{i=1}^n (x_{i,j} - \bar{x}_j)^2 \right]^{\frac{1}{2}} \quad (60)$$

$$\bar{x}_j = \frac{1}{n} \sum_{i=1}^n x_{i,j} \quad (61)$$

where  $\mathbf{1}_p$  is a  $p$  dimensional vector of  $\mathbf{1}$ s. After extensive testing of our simulation routine with the 10 indexes included in the demonstration set, we concluded that our choice of  $\mathbf{h}$  needed to be scaled down by a factor of 10. We provide visual justification for this heuristic in [Figure 5](#) and [Figure 6](#).

In the top half of [Figure 5](#), the density for the sample distribution of SMC (colored blue) is plotted using rolling 21 day intervals from the observed history of the S&P 500 restricted to where the compound return for the period was between 1% and 2% ( $0.01 \leq r \leq 0.02$ ). In the bottom half, the densities for the sample distribution of SMC are plotted using simulated data with (Adjustment Factor = 10, colored red) and without (Adjustment Factor = 1, colored green) scaling down the bandwidth. For the simulations, the period return constraint was set to 1.5% ( $x_0 = \log(1 + 0.015)$ ), the midpoint of the data interval used to generate the density



**Figure 5.** 21 Day SMC densities for SPX.

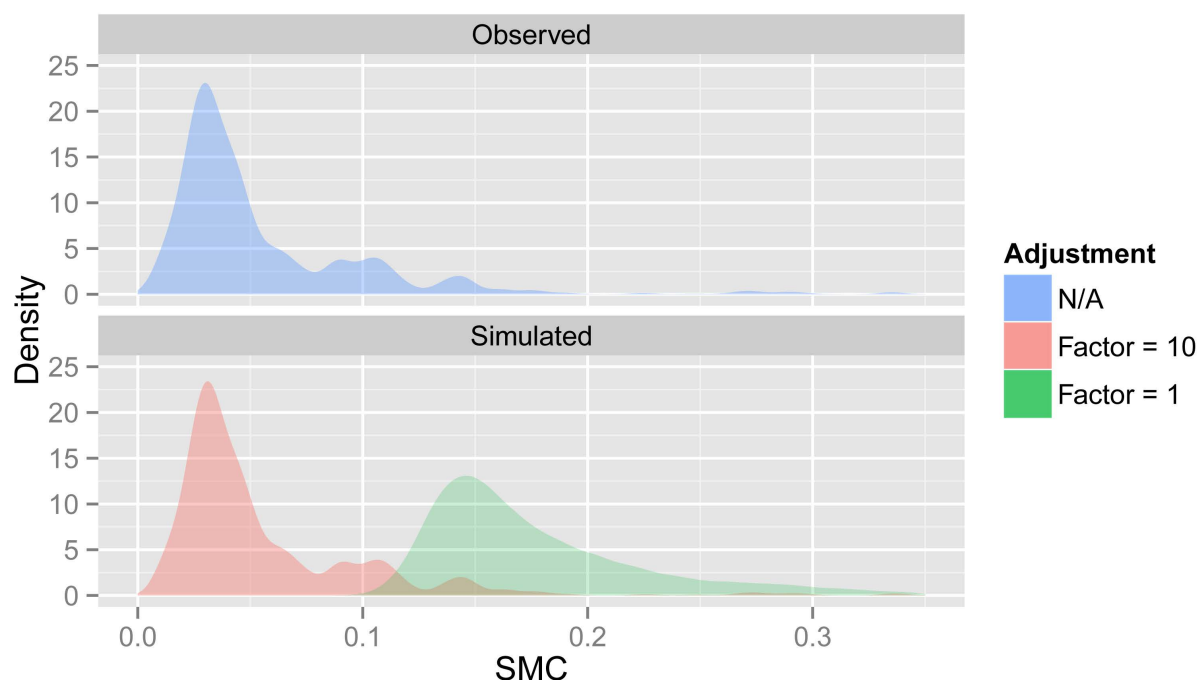


Figure 6. 252 day SMC densities for SPX.

in the top half of the plot. Figure 6 follows the same layout and color scheme as Figure 5, but this time using rolling 252 day intervals and different target returns. Only periods where the compound return on the S&P 500 was between 10% and 15% ( $0.1 \geq r \geq 0.15$ ) were used in the top half, and the return constraint for simulated data in the lower half was 12.5% ( $x_0 = \log(1 + 0.125)$ ). For all SMC calculations (using observed or simulated data), a leverage multiple of +3 is included as if we were approximating LETF Ticker SPXL, where

$(R_j = 3R_{SPX,j} - \text{MgmtFee}_{SPXL})$  is the series of daily returns used for testing. Shortfall from Maximum Convexity was used as the test statistic to stay consistent with the rest of our work, but an adjustment factor of 10 appeared equally appropriate when standard deviation was used.

In the figures shown (and including many other cases not shown), it was obvious to us that the bandwidths derived from our simplified Scott Rule were producing simulated data that looked exaggerated when plotted against the comparable subset of observed data. We decided on a final scaling factor of 10 through trial and error, but we think the results shown in Figure 5 and Figure 6 speak for themselves. Our final bandwidth

selection used in the following application examples is then  $\hat{h} = 0.1 * \left( \frac{1}{p} \sum_{j=1}^p s_j \right) * n^{-1/(p+4)} * \mathbf{1}_p$ .

## 5. Case Study: Simulating LETF Returns with Underlying Index Return Constraints and Calculating Nonparametric $p$ -values for Volatility Statistics

We conclude with a demonstration of our three contributions in one example. Our intention is to capture the significance, or rarity, of the volatility experienced during the chosen period as measured by Shortfall from Maximum Convexity. A  $p$ -value for the observed SMC value will be calculated with respect to a SMC sample distribution generated by our LETF return model. In the experimental setting described here a  $p$ -value of 1 means the lowest possible volatility was observed (if the period in question was repeated, greater volatility would be experienced 100% of the time), and 0 means the highest possible volatility was observed (if the period in question was repeated, greater volatility would be experienced 0% of the time).

The absolute low point of the U.S. equity markets during the 2008-2009 recession occurred in early March 2009, so we will analyze the realized volatility for that month (22 days) as experienced by LETF Ticker TNA. The underlying index, the Russell 2000, returned 8.92% for the month (the market low point was early in the

month, and the rebound was underway by month end). The log form of this value is used as the input total period return of our constrained simulation routine from Section 4 to sample 100,000 sequences of 22 days of Russell 2000 returns. For each of the 100,000 index return samples, our tracking error simulation routine is executed to sample sequences of TNA tracking errors. The data are combined with a +3 leverage multiple (and daily management fee) to produce a final sample of 22 day TNA returns sequences.

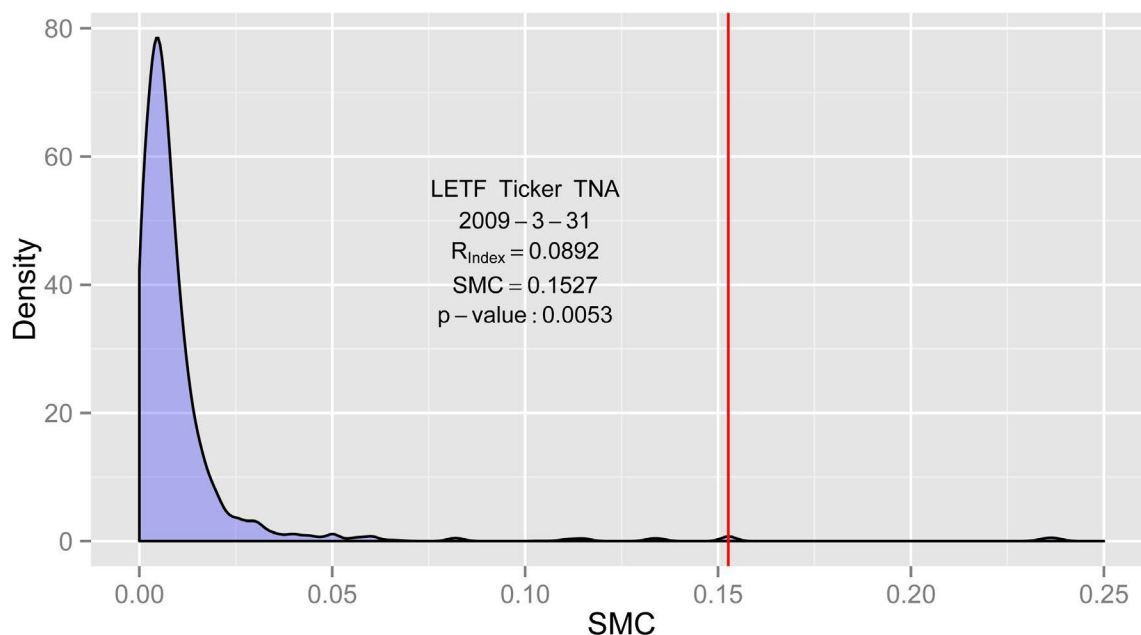
As displayed by the row for Ticker TNA in [Table 1](#), the lag parameter  $\ell$  for the simulations was set to 3. The history of daily returns for the Russell 2000 index dating back to 1/2/1979 was arranged into a data matrix of 9151 observations of rolling 25 day intervals (22 days plus 3 lags) to derive the kernel density estimate for constrained simulation. The history of daily returns for the LETF Ticker TNA dating back to 11/11/2008 was combined with the matching Russell 2000 daily returns into a data matrix of 1536 observations of 8 dimensions (4 day sequence of index returns, 4 day sequence of LETF tracking errors) to derive the kernel density estimate for tracking error simulation. All observed values were log transformed before simulation, and our bandwidth selections described in Sections 3 and 4 were applied accordingly.

Our realized volatility statistic SMC was calculated for each 22 day sample and the density of the final SMC sampling distribution is plotted below in [Figure 7](#). The red line marks the actual SMC value of 0.1527 that was observed for March 2009. The resulting  $p$ -value for this observation with respect to the simulated sample is 0.0053, which indicates that the proportion of simulated SMC values that were less than or equal to the observed value is 0.9947.

Our constrained simulation routine for index returns guarantees all the 22 day index return samples compound to the observed index return of 8.92%. Given this fact, the range of outcomes represented by the SMC sample is entirely the result of index return volatility and tracking error. Based on the output of our simulation methods we conclude that the realized volatility for TNA in March 2009 was greater than the overwhelming majority of outcomes that may be observed while the Russell 2000 experiences an 8.92% increase over the same period.

## 6. Conclusions

Leveraged ETF returns are unique due to the dynamics produced by daily compounding, and we hope to have illuminated some of the challenges involved with modeling these returns over time horizons longer than one day. We have presented a nonparametric framework for modeling daily LETF returns that feature a simulation method for LETF tracking errors and a method for simulating index returns with a constrained period return.



**Figure 7.** Simulated 22 day SMC density for LETF ticker TNA, with all underlying index returns sequences compounding to 8.92%.



The two methods allow for incorporating additional index returns data and controlling for index performance, respectively. We feel that these contributions can improve the public understanding of the relationship between LETF return volatility and index returns.

Our methods can model almost any dependence structure that may exist in LETF returns data and each can be used separately if desired. When used together, the two methods complement each other nicely, enabling powerfully customized risk summaries of LETF investing strategies. Finally, the constrained simulation method is not limited to our particular index returns dataset. It can be used to sample any generic multivariate data with a constraint on the sum of the random vector elements.

## Acknowledgements

We thank Dan Cohen, the Editor, and an anonymous reviewer for their helpful comments.

## References

- [1] Fisher, D. (2014) Leverage Your Way to a Richer Retirement. *Forbes*.
- [2] Scott, J.S. and Watson, J.G. (2013) The Floor-Leverage Rule for Retirement. *Financial Analysts Journal*, **69**, 45-60. <http://dx.doi.org/10.2469/faj.v69.n5.2>
- [3] Simpson, S.D. (2015) A Brief History of Exchange-Traded Funds. *Investopedia*. <http://www.investopedia.com/articles/exchangetradedfunds/12/brief-history-exchange-traded-funds.asp>
- [4] Yates, T. (2007) Dissecting Leveraged ETF Returns. *Investopedia*. <http://www.investopedia.com/articles/exchangetradedfunds/07/leveraged-etf.asp>
- [5] Jarrow, R.A. (2010) Understanding the Risk of Leveraged ETFs. *Finance Research Letters*, **7**, 135-139. <http://dx.doi.org/10.1016/j.frl.2010.04.001>
- [6] Avellaneda, M. and Zhang, S. (2010) Path-Dependence of Leveraged ETF Returns. *SIAM Journal on Financial Mathematics*, **1**, 586-603.
- [7] Lu, L., Wang, J. and Zhang, G. (2012) Long Term Performance of Leveraged ETFs. *Financial Services Review*, **21**, 63-80.
- [8] Dobi, D. and Avellaneda, M. (2012) Structural Slippage of Leveraged ETFs. <http://ssrn.com/abstract=2127738>
- [9] Tuzun, T. (2012) Are Leveraged and Inverse ETFs the New Portfolio Insurers? Technical Report, Fed. Reserve Board, Div. of Research and Stat. and Monetary Affairs, Washington DC.
- [10] Cooper, T. (2010) Alpha Generation and Risk Smoothing Using Managed Volatility. <http://ssrn.com/abstract=1664823> <http://dx.doi.org/10.2139/ssrn.1664823>
- [11] Bouchey, P. and Nemtchinov, V. (2013) BENDING CAPM: Why Do High Volatility Stocks Underperform? Technical Report, Parametric Portfolio.
- [12] Direxion Funds (2015) Direxion Daily S&P 500 Bull and Bear 3x Shares Fact Sheet. <http://www.direxioninvestments.com/wp-content/uploads/2014/02/SPXLSPXS-Fact-Sheet.pdf>
- [13] Direxion Funds (2014) Direxion Daily S&P 500 Bull 3x Prospectus. <http://direxioninvestments.onlineprospectus.net/DirexionInvestments/SPXL/index.html>
- [14] ETF Database (2014) Definitive List of Leveraged Equities ETFs. <http://etfdb.com/etfdb-category/leveraged-equities>
- [15] Direxion Funds (2014) Direxion Daily Bull 3x and Bear 3x ETFs. <http://www.direxioninvestments.com/etfs>
- [16] ProShares (2014) ProShares Products Leveraged 3x and Inverse 3x ETFs. <http://www.proshares.com/funds>
- [17] Bloomberg (2014) Bloomberg Professional [Online]. Subscription Service.
- [18] Ginley, M. (2015) Shortfall from Maximum Convexity. <http://arxiv.org/abs/1510.00941>
- [19] Markowitz, H. (1952) Portfolio Selection. *The Journal of Finance*, **7**, 77-91. <http://www.jstor.org/stable/2975974>
- [20] Conover, W.J. (1971) Practical Nonparametric Statistics. 3rd Edition, Wiley, Hoboken.
- [21] Scott, D.W. (1992) Multivariate Density Estimation: Theory, Practice, and Visualization. Wiley, Hoboken. <http://dx.doi.org/10.1002/9780470316849>
- [22] Mardia, K.V., Kent, J.T. and Bibby, J.M. (1979) Multivariate Analysis. Academic Press, London.

## Appendix

**Chart 1.** Demonstration set of selected pairs of LETFs.

Index	Issuer	Leverage	LETF	LETF Name	LETF Start	Obs.
DJUSFN	ProShares	+3	FINU	UltraPro Financials	2012-07-13	596
DJUSFN	ProShares	−3	FINZ	UltraPro Short Financials	2012-07-13	391
GDM	Direxion	+3	NUGT	Daily Gold Miners Bull 3x Shares	2010-12-09	1013
GDM	Direxion	−3	DUST	Daily Gold Miners Bear 3x Shares	2010-12-09	1013
INDU	ProShares	+3	UDOW	UltraPro Dow30	2010-02-12	1221
INDU	ProShares	−3	SDOW	UltraPro Short Dow30	2010-02-12	1221
IXT	Direxion	+3	TECL	Daily Technology Bull 3x Shares	2008-12-18	1510
IXT	Direxion	−3	TECS	Daily Technology Bear 3x Shares	2008-12-18	1510
MVRSX	Direxion	+3	RUSL	Daily Russia Bull 3x Shares	2011-05-26	897
MVRSX	Direxion	−3	RUSS	Daily Russia Bear 3x Shares	2011-05-26	886
MXEA	Direxion	+3	DZK	Daily Developed Markets Bull 3x Shares	2008-12-18	1510
MXEA	Direxion	−3	DPK	Daily Developed Markets Bear 3x Shares	2008-12-18	1509
MXEF	Direxion	+3	EDC	Daily Emerging Markets Bull 3x Shares	2008-12-18	1510
MXEF	Direxion	−3	EDZ	Daily Emerging Markets Bear 3x Shares	2008-12-18	1510
NDX	ProShares	+3	TQQQ	UltraPro QQQ	2010-02-12	1221
NDX	ProShares	−3	SQQQ	UltraPro Short QQQ	2010-02-12	1221
RTY	Direxion	+3	TNA	Daily Small Cap Bull 3x Shares	2008-11-06	1539
RTY	Direxion	−3	TZA	Daily Small Cap Bear 3x Shares	2008-11-06	1539
SPX	Direxion	+3	SPXL	Daily S&P 500 Bull 3x Shares	2008-11-06	1539
SPX	Direxion	−3	SPXS	Daily S&P 500 Bear 3x Shares	2008-11-06	1539

**Chart 2.** Underlying indexes for demonstration set of LETFs.

Index	Index Name	Index Start	Obs.
DJUSFN	Dow Jones U.S. Financials	1992-01-02	5891
GDM	NYSE Arca Gold Miners	1993-09-20	5352
INDU	Dow Jones Industrial Average	1928-01-04	21,943
IXT	S&P Technology Select Sector	1998-01-02	4271
MVRSX	Market Vectors Russia	2007-10-01	1869
MXEA	MSCI EAFE	1971-12-31	11,206
MXEF	MSCI Emerging Markets	1988-01-01	7032
NDX	NASDAQ-100	1985-02-05	7532
RTY	Russell 2000	1979-01-02	9175
SPX	S&P 500	1928-01-03	21,842