Simulation of Leveraged ETF Volatility Using Nonparametric Density Estimation

Matthew Ginley

May 20th, 2016
Russell 2000 (RTY), +1x ETF (IWM), +3x ETF (TNA), 10/15/15 - 4/15/16

\[ R_{IWM} = -2.7\% \]
\[ R_{RTY} = -2.74\% \]
\[ R_{TNA} = -12.52\% \]
Talk Outline

1. Realized Volatility of LETF Returns
2. Simulation Methods based on Kernel Density Estimators
3. KDE Application Example: LETF Options Strategy
Leveraged ETFs (LETFs)

- First appeared in 2006. Now offered on many underlying indexes.
- Leverage: LETF offers $\beta$ times the index return.
- Typically, $\beta \in \{+2, -2, +3, -3\}$. Greater values create potential for more extreme results.
- Compounding: Leveraged return is calculated daily, so every day’s return is compounded with leverage.
- Daily compounding of leveraged returns often times results in underperformance, or volatility drag.
<table>
<thead>
<tr>
<th>Day</th>
<th>Index Value</th>
<th>LETF Value</th>
<th>Index Return</th>
<th>LETF Return</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>100</td>
<td>1%</td>
<td>3%</td>
</tr>
<tr>
<td>2</td>
<td>101</td>
<td>103</td>
<td>1%</td>
<td>3%</td>
</tr>
<tr>
<td>3</td>
<td>102.01</td>
<td>106.09</td>
<td>-1%</td>
<td>-3%</td>
</tr>
<tr>
<td>4</td>
<td>100.9899</td>
<td>102.9073</td>
<td>-1%</td>
<td>-3%</td>
</tr>
<tr>
<td>5</td>
<td>99.98</td>
<td>99.82008</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Volatility Drag

For $p$ day vectors of returns, we assume ...

$$R_{LETF} = \beta \cdot R_{Index} - \text{Fee}_{MGMT} + \epsilon$$

$$\prod(1 + R_{LETF}) - 1 \approx \prod(1 + \beta \cdot R_{Index}) - 1$$

As realized volatility increases ...

$$\prod(1 + \beta \cdot R_{Index}) - 1 \leq \beta \cdot [\prod(1 + R_{Index}) - 1]$$

As realized volatility decreases, inequality flips direction. “gains on gains"
Rolling 62 Day Returns, +1x ETF vs. +3x ETF, 11/05/08 - 4/15/16
Density for Multi-day Returns with Period Return Constraint

Assume ...

\[ \log(1 + R) = X \sim f(x), \] density function for \( p \) dimensional log returns

We want an estimator for ...

\[ f(x|\sum x < c) \]

We will fit and then transform a Kernel Density Estimator for \( X \)
Histogram of 3 points (-1.6, -0.6, 1.6), binwidth = 1
KDE of 3 points (-1.6, -0.6, 1.6), Gaussian Kernel, bandwidth = 1
Example KDE of 2 Day log returns
How to Transform the KDE

\[ f(x) \approx \hat{f}(x), \text{ kernel density estimate} \]

\[ \hat{f}(x) \Rightarrow \hat{f}(x - p, \sum x) \]

\[ \hat{f}(x - p, \sum x) \Rightarrow \hat{f}(x - p | \sum x) \hat{f}(\sum x) \]

\[ \hat{f}(\sum x) \Rightarrow \hat{f}(\sum x | \sum x < c) \]

\[ \hat{f}(x - p | \sum x) \hat{f}(\sum x | \sum x < c) \approx f(x | \sum x < c) \]
Code Snippet of Main Simulation Function

```r
> str(gross_val$RTY)
  Named num [1:9508] 0.00716 0.01789 0.01492 0.01328 ...
  - attr(*, "names")= chr [1:9509] "19790102" "19790103" "19790104" ...
> R_mat <- do_findatamatroll(gross_val$RTY, 62)
> str(R_mat)
  num [1:9447, 1:62] 0.00716 0.01789 0.01492 0.01328 ...
  - attr(*, "dimnames")=List of 2
    ..$: chr [1:9447] "19790328" "19790329" "19790330" "19790402" ...
    ..$: chr [1:62] "1" "2" "3" "4" ...
> R_sim <- sim_index(R_mat, k = 100000, r_0 = 0.1, r_ab = Inf)
> str(R_sim)
  num [1:100000, 1:62] 2.16e-02 7.21e-02 8.18e-03 7.65e-03 -6.34e-05 ...
  - attr(*, "dimnames")=List of 2
    ..$: chr [1:100000] "19990105" "20090605" "19830523" "19910129" ...
    ..$: chr [1:62] "1" "2" "3" "4" ...
> mean(apply(1 + R_sim, 1, prod) - 1 < 0.1)
[1] 0
> mean(apply(1 + R_sim, 1, prod) - 1 >= 0.1)
[1] 1
```
Key References

  Proposes constrained KDE and tracking error methods (not restricted to finance/time series), and LETF volatility example.

- “Shortfall from Maximum Convexity", arXiv.org, November 2015
  Appendix/Companion with more insights on realized volatility of LETF returns and volatility drag.

- https://github.com/bonzodroog, mginley@rice.edu
  All code for simulation methods, presentation visuals, and Excel templates for Bloomberg data.

- Dr. Tim Leung, faculty at Columbia University, LETF Expert
  Continuous time models of LETF returns, and a whole lot more.
Hypothetical +1x ETF vs. +3x LETF Returns

The graph illustrates the hypothetical returns of +1x ETF vs. +3x LETF under different market conditions. The x-axis represents the +1x ETF Return (%), while the y-axis shows the +3x ETF Return (%). The graph includes a line for +3x Call Strike, another for +1x Call Strike, and a vertical line indicating +3x LETF MAX. The data points demonstrate that as the +1x ETF Return increases, the +3x ETF Return also increases, with a cap at +3x LETF MAX.
write 1 contract on 3x BULL CALL
buy 3 contracts on (1x BULL CALL or 1x BEAR PUT)

\[ V = 3 \times V_{1xBULL} - V_{3xBULL} \]

\[ V = 3 \times (Payout_{1xBULL} - Premium_{1xBULL}) - (Payout_{3xBULL} - Premium_{3xBULL}) \]

\[ V = (3 \times Payout_{1xBULL} - Payout_{3xBULL}) + (Premium_{3xBULL} - 3 \times Premium_{1xBULL}) \]

\[ V = Payout_{NET} + Premium_{NET} \]
Expected Value of Strategy

\[ E[V] = E[Payout_{NET}] + \text{Premium}_{NET} \]

\[ R_{1xBULLStrike} = \left( \frac{\text{OptionStrike}_{1xBULL}}{\text{UnderlyingClose}_{1xBULL}} \right) - 1 \]

\[ E[V] = E[Payout_{NET} \mid R_{Index} < R_{1xBULLStrike}] \cdot P[R_{Index} < R_{1xBULLStrike}] \\
+ E[Payout_{NET} \mid R_{Index} > R_{1xBULLStrike}] \cdot P[R_{Index} > R_{1xBULLStrike}] \\
+ \text{Premium}_{NET} \]

\[ E[V] = E[Payout_{NET} \mid R_{Index} > R_{1xBULLStrike}] \cdot P[R_{Index} > R_{1xBULLStrike}] + \text{Premium}_{NET} \]
1. Observe index, ETFs, and ETF Option prices for a given index and expiration date

2. For each 1xBULL strike, simulate index returns distributions with appropriate constraints, and then derive 3xBULL ETF returns distributions

3. Using simulated data, derive net payout distributions for each option pair

4. Review other information as necessary to arrive at a formal trading rule (e.g. size of initial investment, or net premiums).
Backtest Details

- Analyze Russell 2000, 1xBULL ETF IWM, 3xBULL ETF TNA, 4/15/16 Expiry, observe pricing data as of market close on 1/15/16
- IWM closed at 100.12, TNA closed at 44.08
- 25 IWM call strikes traded (90 - 118), or implied returns (-10% to +18%)
- 26 TNA call strikes traded (30 - 80), or implied returns (-32% to +82%)
- 650 possible pairs, of which 391 satisfy our requirement
- As of close on 4/15/16, IWM reached 112.45 (12.31%), TNA reached 60.66 (37.64%)
Actual Profit vs. Mean Forecast Profit as of 1/15/2016

Result
- Red: Loser
- Black: Winner

Actual Profit ($) vs. Mean Forecast Profit ($)
KDEs are fast, flexible, and simple. Let the data speak for themselves.

KDEs are a useful tool for analyzing LETFs because of their ease of implementation and abundance of Index return data available.

R and R Studio make the brainstorming and development process more productive and enjoyable.

Thanks to R/Finance 2016 and Rice University!

Special thanks to R Studio and Rishi Narang!