Memorandum

To: File

From: The Division of Economic and Risk Analysis (DERA)[1]

Date: November 16, 2017

Re: Inferring Non-Transparent ETF Portfolio Holdings

---

1 Specifying the reverse engineering problem

This document discusses the technical aspects of the extent to which non-transparent ETF holdings can be reverse engineered. In this section, we introduce notation to precisely specify the problem and to better understand the regressions used in the various analyses discussed below. Consider a universe of $i = 1, \ldots, N$ stocks with mid-prices P_i^t at each time $t = 0, \ldots, T$ within a trading day ($t = 0$ is the market open and $t = T$ is the market close). If a fund owns n_i shares of each stock as of the prior days close, its value at time t is $P_F^t = \sum_{i=1}^N n_i P_i^t$. Precidian proposes to rescale this value periodically such that its initial price is approximately \$20 at the start of each day ($t = 0$), producing a verified intra-day indicative value (VIIV) at each time t of

$$VIIV^t = \frac{20}{P_F^0} \cdot P_F^t = 20 \frac{\sum_{i=1}^N n_i P_i^t}{P_F^0} = \sum_{i=1}^N w_i \frac{20}{P_i^0} P_i^t$$

where $w_i = n_i P_i^0 / P_F^0$ are the portfolio weights in each of the underlying assets at $t = 0$.[2] Without rounding, the portfolio weights w_i can be exactly recovered from N observations of the *scaled* asset prices $\frac{20}{P_i^0} P_i^t$ and $VIIV_t$ as long as the asset prices are linearly independent. While stock price movements tend to be highly correlated, they are not perfectly linearly dependent, and in simulations of random portfolios we are able to solve this exact linear algebra problem without the need to use a statistical estimation technique such as ordinary least squares (OLS) regression.

Precidian's claim is that rounding the VIIV to the nearest penny prevents the portfolio weights w_i from being reverse engineered. If we denote the rounded VIIV at time t as $\overline{VIIV}^t$, it can be expressed as

$$\overline{VIIV^t} = VIIV^t + (\overline{VIIV^t} - VIIV^t) = \sum_{i=1}^N w_i \frac{20}{P_i^0} P_i^t + \epsilon_t$$

[1] This is a memorandum by the staff of the U.S. Securities and Exchange Commission. The Commission has expressed no view regarding the analysis, findings, or recommendations contained herein.

[2] $t = 0$ can equivalently represent the market close on the prior day, it does not change the analysis.

where $\epsilon_t = \overline{VIIV^t} - VIIV^t$ is the rounding error.[3] We are interested in the extent to which the weights w_i or the number of shares n_i can be reverse engineered by observing the rounded quantity $\overline{VIIV^t}$ and the un-rounded mid-prices P_i^t.

2 Existing Analyses

Several relevant studies attempt to assess the extent to which portfolio holdings can be reverse engineered: 1) Staff from the Risk and Examination Office (REO) in the Division of Investment Management (IM) performed an initial analysis of whether weights could be reverse engineered; 1) Precidian responded to some initial feedback from IM on the issue with a study by Ricky Cooper that uses OLS regressions to claim that reverse engineering is not possible once the VIIV is scaled and rounded to the nearest penny; 2) Precidian submitted an additional study by Lawrence Glosten that uses a more complex estimation technique, ("LASSO" regression) and reaches a similar conclusion; 3) Blue Tractor recently submitted an analysis that shows holdings *can* be reverse engineered to a high degree of accuracy for a small universe of 100 stocks. In this section, we examine each of these analyses and their conclusions.

2.1 REO analysis

REO performed an analysis for several dates using the NASDAQ 100 as a universe from which to construct portfolios with true weights w_i on each stock, where both long and short positions were allowed.[4] They scale these portfolios to have an initial value of \$20 on each trading day, and run regressions using both the full precision price and a rounded (to the penny) price. Using mid-prices observed at 10 second intervals, they run the regressions

$$R^t_{VIIV} = \sum_{i=1}^{N} \beta_i R^t_i + \epsilon_t \qquad R^t_{\overline{VIIV}} = \sum_{i=1}^{N} \beta_i R^t_i + \epsilon_t$$

fitting both the exact and rounded VIIV *returns* on the individual asset returns implied by the mid-price movements, where these returns are defined by

$$R^t_{VIIV} = \frac{VIIV^t}{VIIV^{t-1}} \qquad R^t_{\overline{VIIV}} = \frac{\overline{VIIV}^t}{\overline{VIIV}^{t-1}} \qquad R^t_i = \frac{P^t_i}{P^{t-1}_i}$$

The estimated coefficients β_i are intended to be portfolio weights, so they are constrained to sum to 1: $\sum_{i=1}^{N} \beta_i = 1$.

Note that the expressions for the $VIIV$ derived in the previous section uses the *levels* of the variables while this analysis uses *returns*. If the problem is specified in terms of *returns*,

[3]Note that, technically, the standard regression assumption that the error is independent of the regressors doesn't apply here: the error term is *exclusively* a function of the regressors, albeit a complicated one.

[4]The code from this analysis formed the basis for the our additional simulations discussed in the next section.

it can be shown that the true relationship between returns on the VIIV portfolio and the returns of its constituents is

$$R^t_{VIIV} = \sum_{i=1}^{N} w_i^{t-1} R_i^t \qquad w_i^{t-1} = \frac{n_i P_i^{t-1}}{\sum_{i=1}^{N} n_i P_i^{t-1}}$$

In other words, when returns are used, there is not a constant relationship between the VIIV return and the individual asset returns: it is based on the weights as of the end of the previous time period, and these weights fluctuate throughout the day. In practice, this means that estimating constant regression coefficients β_i captures the average portfolio weights over the course of the day, which, if price fluctuations are random, is likely to result in estimates of the initial weights w_i that are close to the estimates produced by using the price level specification in the previous section. Over multiple days, weight estimates based on returns will under-perform estimates based on price levels.

2.2 Ricky Cooper's Analysis

For stocks in the NASDAQ 100, Cooper constructs random portfolios by selecting 40% of the universe to have non-zero weights and equally weighting across these stocks (so each has a weight of 0.025). He uses these "weights" as the number of shares in forming the fund price, i.e.

$$P_F^t = \sum_{i=1}^{N} w_i P_i^t$$

In other words, the w_i Cooper is estimating are not portfolio weights but rather the fractional number of shares held in each stock: the same number of shares is held in each stock, but the dollar amounts invested in each stock differ. The above price weighting implies that the unscaled portfolio values P_F^t are on the order of \$100. Cooper runs three regressions to estimate w_i, each using a day of one-second observations, and repeats the analysis for 44 dates. Results are measured using the aggregate mean square error (MSE) of the fitted weights relative to the true weights, which can be interpreted as the standard error of the fitted weights:[5]

1. As a benchmark, before rounding and scaling, Cooper regresses the exact fund price on the exact asset prices:

   $$P_F^t = \sum_{i=1}^{N} w_i P_i^t + \epsilon_t$$

   The estimated weights closely match the constituents (the mean square error (MSE) between the true and estimated weights is 0).[6] This result is expected: as mentioned above, with exact fund prices, the weights can be solved for exactly using only N

[5] For T days, with N fitted regression weights $\hat{w}_{i,t}$ on each day t, the MSE is given by $\sqrt{\frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}(\hat{w}_{i,t} - w_{i,t})^2}$

[6] He actually uses the last *traded* price on both the left and right hand sides instead of mid-prices, but this doesn't make any difference in the exact case.

observations, and OLS regressions using $T > N$ data points should achieve similar results.

2. Cooper runs a similar regression with the exception that the fund price $\overline{P}_F^t$ is rounded to the nearest penny, and obtains a reasonable fit: the MSE of his regression weights versus the true weights is 0.0012, which is an order of magnitude smaller than the average naive portfolio weight of 0.01 (there are 100 stocks and a naive portfolio assignment would choose equal weights for all stocks).

3. Finally, Cooper scales the fund price so that its initial price is \$20, and obtains an MSE of 0.0104, which is similar to the MSE of a naive equally weighted portfolio across all 100 stocks (0.0122). Cooper claims that this demonstrates that, once the fund price is scaled to \$20, one cannot do much better than picking a naive portfolio. However, the result is actually due to not fully accounting for the effects of scaling. Based on the above fund price P_F^t in Cooper's analysis, the scaled/rounded VIIV, similar to the one derived in the previous section, is

$$\overline{VIIV}^t = \sum_{i=1}^{N} w_i \frac{20}{P_F^0} P_i^t + \epsilon_t$$

where ϵ_t is the rounding error. That means that, to recover the weights w_i as closely as possible, the rounded and scaled VIIV should be regressed on *scaled* prices $\frac{20}{P_F^0} P_i^t$. Because Cooper only regresses on the *unscaled* prices P_i^t, the best case scenario, under which the regression is estimated with no error, is that the estimates $\hat{w}_i$ are related to the true weights by

$$\hat{w}_i = \frac{20}{P_F^0} w_i$$

If a similar error-free regression of the VIIV on properly scaled prices were run, the portfolio weights would be perfectly recovered ($\hat{w}_i = w_i, MSE = 0$). For Cooper's setup, where $M < N$ stocks are chosen from the universe and equally weighted, it can be shown that the best case MSE if the regression incorrectly uses *unscaled* prices is

$$MSE(\hat{w}_i = \frac{20}{P_F^0} w_i) = \sqrt{\frac{1}{MN}\left(\frac{20}{P_F^0} - 1\right)}^2$$

Cooper claims that the unscaled ETF price P_F^0 is on the order of \$100, and his setup has $N = 100$ and $M = 40$, implying an approximate best case $MSE \approx 0.0126$ according to the above equation. This is the same order of magnitude as the MSE claimed by Cooper for the rounded and scaled VIIV: the effect of incorrectly handling scaling issues explains the change in magnitude of the MSE introduced in Cooper's third regression.

In sum, Cooper concludes that rounded (but unscaled) VIIV's can be used to obtain a decent approximation of the portfolio constituents from a small, 100 stock universe. While scaling the VIIV does reduce the precision of this approximation, it does not eliminate it (as claimed by Cooper). More generally, with all of these studies, translating low MSE, high R^2, or high correlation estimates of an ETF's portfolio weights into a profitable trading strategy would require more explicit assumptions about a fund's alpha generating process. It would not necessarily require the exact replication of a portfolio. For example, effectively inferring the *ranking* of the portfolio's weights is potentially informative about which stocks an active ETF will be buying/selling.

2.3 Lawrence Glosten's Analysis

Glosten's analysis differs from Cooper's in two key respects: 1) he uses a larger universe (the Russell 1000) from which to construct random portfolios using a randomly chosen subset of 130 stocks; 2) he primarily estimates portfolio weights using the "LASSO" regression technique instead of OLS.

Glosten uses LASSO because it is a "sparse regression technique" that is designed for problems where weights on some of the variables are zero.[7] OLS will also infer the correct weights with enough data, while LASSO makes a bias-variance trade-off: with limited data, LASSO may result in better prediction errors for the independent variable even though it introduces bias into coefficient estimates. However, for the purpose of reverse engineering an ETF's portfolio holdings, small prediction errors are not the primary objective. While the sparse nature of the LASSO algorithm's output makes intuitive sense in situations where a fund only invests in a subset of the stocks in its universe, it is not obvious a priori that it should outperform OLS.

Glosten runs his analyses on portfolios constructed from a random number of shares scaled so that the fund price is in the \$20 to \$60 range, performing the LASSO regression

$$\overline{VIIV}^t = \sum_{i=1}^{N} \beta_i P_i^t + \epsilon_t$$

Because the regressors are not scaled, the regression produces estimates of the *number of shares of each security* held. While the regressions have high R^2 values with respect to how well the model fits the VIIV to the mid-prices, the overlap between the names held in the estimated versus true portfolio is small. He also runs several OLS regressions as a benchmark and obtains similar results.

Glosten's study does not include details on how the LASSO algorithm was implemented, making replication of the results difficult. In our experience, LASSO algorithm results can vary with the software package used and various options related to how data is standardized before the model is estimated. Therefore, it is possible that other LASSO implementations could achieve more accurate estimates of portfolio holdings. Finally, Glosten's finding that OLS and LASSO are unable to accurately recover portfolio holdings for a larger universe of available portfolio securities (the Russell 1000) does not rule out the possibility that other techniques might perform better on stock universes of the same size.

[7] In addition to the typical least-squares-error objective of OLS, LASSO imposes a penalty on the sum of the absolute coefficient values, encouraging some of them to be zeroed out.

2.4 Blue Tractor's Analysis

Blue Tractor provided an analysis by Anthony Hayter to counter Precidian's claim that the portfolio holdings of a non-transparent ETF cannot be reverse engineered. Hayter does not use actual pricing data: instead, he measures the correlation and volatility of 10 Nasdaq stocks, and uses a single "average" correlation value across the simulated stock universe. Hayter estimates the correct regression specification in levels (see Section 1), dividing by the initial value $VIIV^0 = \$20$ to estimate

$$\frac{\overline{VIIV}^t}{20} = \sum_{i=1}^{N} w_i \frac{P_i^t}{P_i^0} + \epsilon_t$$

Hayter forms equally-weighted portfolios from a randomly selected subset of 40% of a universe of 50 and 100 stocks, and simulates data in 15 second increments. Hayter uses a two-pass approach: the above regression is estimated, any stocks with statistically insignificant coefficients are removed from the universe, and the regression is re-estimated.

Using different measures of fit, Hayter shows that with 5 days of data, he can reverse engineer portfolios in this small universe to a very high degree of accuracy: his weights are only off by an average of 0.000776 relative to the true portfolio weight of 0.025, exhibit very few false-positives, and never exclude a stock that is in the true portfolio. With only one day of data and 15 second VIIV increments, the results are less accurate, but this is much less data than the one second increments that Precidian's ETF structure would provide. On a larger universe with 1 second data, the results would probably lie somewhere in between these two extremes. The main caveat that applies to Hayter's analysis is that the simulated data has a very simple correlation structure that may understate the difficulty involved in estimating the model from actual prices. Nonetheless, the two-pass aspect of his regression approach is an example of an alternative estimation technique that potentially improves upon the standard OLS and LASSO estimations above.

3 DERA Simulations

DERA performed additional simulations by extending code developed for the REO analysis described above using 10 second intervals. Random portfolios were generated using both equally weighted and randomly weighted subsets of a stock universe, for different size universes (the DOW 30, NASDAQ 100, S&P 500, and Russell 1000). As in the REO analysis, DERA initially worked with returns to the unrounded and rounded VIIV and regressed them on the returns implied by mid-prices, running the regressions

$$R_{VIIV}^t = \sum_{i=1}^{N} \beta_i R_i^t + \epsilon_t \qquad R_{\overline{VIIV}}^t = \sum_{i=1}^{N} \beta_i R_i^t + \epsilon_t$$

which means that the resulting estimates β_i are an average of the weights over the course of the day. Any success in reverse-engineering the portfolio weights should therefore be considered a lower bound on what is possible. In practice, this approach is not much different than using the correct level specification in Section 1 for one-day of data, but it increasingly under performs as the number of days used to estimate the model increases. The primary reason for using returns instead of price levels, despite this mis-specification, is that it was the only specification that worked reliably when applying the LASSO algorithm.[8]

DERA's results using both OLS and LASSO to estimate the above return specification:

- Consistent with the above mathematical analysis, the portfolio weights are fully recovered when the VIIV is not rounded, regardless of whether OLS, LASSO, or linear algebra is used. This holds even when the universe size is large.

- Once the VIIV is rounded, for both OLS and LASSO the effectiveness of reverse engineering decreases as the size of the universe increases. Table 1 shows this trend using the cross-sectional R^2, which measures how well the estimated model weights explain the true model weights. Table 3 shows the same trend using the Spearman rank-correlation of the true vs. estimated weights.[9]

- For a given universe size, funds that invest in a larger fraction of the universe are harder to reverse engineer. Tables 2 and 4 demonstrate this, again using the cross-sectional R^2 and Spearman rank-correlations of the true versus estimate weights.

- While it's hard to directly compare our 10 second LASSO results to Glosten's one second results, Table 5 shows that, like his analysis, LASSO produces roughly the same number of false-positives as matches, so a given name in the estimated portfolio has an approximately 50% chance of being in the actual portfolio. This table does not capture the interaction of the magnitudes of the estimated versus true weights: some of the false-positive positions may be small and may not significantly alter other measures of fit like R^2 or rank-correlation.

- Both algorithms *perform better with more observations relative to the universe size.* These analyses used 10 second intervals, so we would expect all of the results to improve with higher frequency observations such as the one second VIIV observations that would be available under the proposed ETF structure. Table 6 demonstrates that if the number of observations is increased by a factor of 50 by using 50 days of data, and if the appropriate regression specification is used (in levels rather than returns), the precision of our weight estimates increases from an R^2 of 7% to an R^2 of 96.7%. We would expect this result to be analogous to using 5 days of 1 second data, assuming no turnover in the portfolio and assuming that intra-day variation in the mid-prices is comparable to the 10-day sample.

[8]DERA used the LASSO implementation in the `Python scikit-learn` library. Many LASSO implementations perform better when input data are standardized, and return data naturally have this feature. When LASSO was used on price levels using the same library, even with various standardization parameters set to `TRUE`, the results were not valid weight estimates. This was the case even when the VIIV was not rounded to the nearest penny, which, as discussed above, should make it easy to recover the portfolio weights.

[9]The Spearman rank correlation is sensitive to infinitesimal noise in estimating the weights. For example, even the exact OLS estimation will produce small non-zero weights for stocks that are not in the portfolio. Removing this estimation noise by setting insignificant weights to zero would make the rank correlation more comparable with the R^2. The rank correlation for the exact estimates in the first two columns of Table 3 can be considered "best case" scenarios for the rank correlation when this noise is not filtered out.

The algorithms described above use some additional information about the problem to constrain estimates. For example, Cooper's estimations impose the constraint that the weights w_i sum to one. DERA was able to improve estimation results using the fact that *each* error term ϵ_t is a result of rounding to the nearest penny, so $\$-0.005 < \epsilon_t < \$0.005$ for all t. By posing the least-squares problem as a quadratic program (QP-OLS), we estimate the correct level specification

$$\overline{VIIV^t} = \sum_{i=1}^{N} w_i \frac{20}{P_i^0} P_i^t + \epsilon_t$$

subject to the constraints that: 1) all weights are positive; 2) all weights sum to 1; 3) all solutions imply a maximum error of \$0.005 *at each observation* t. This imposes an additional $(2T + N + 1)$ constraints relative to the standard OLS algorithm, where N is the number of stocks and T is the number of observation. The results of using this technique include:

- Table 7 shows that if the correct regression specification in levels is used, the QP-OLS technique significantly outperforms standard OLS and, comparing with Table 1, LASSO. For example, using the S&P 500 as the stock universe, the average cross-sectional R^2 using LASSO is approximately 11% for variable-weight portfolios (Table 1, column 4), while QP-OLS achieves an average R^2 of approximately 47% (Table 7, column 4).

- QP-OLS's superior performance is even more apparent for multiple days of data. For the S&P 500 universe and random positive weights, Table 8 shows that QP-OLS can achieve the same degree of accuracy in 5 days of data as achieved by standard OLS in 50 days of data (shown in Table 6), and achieves an R^2 of 0.996 after 10 days. Table 9 shows that, even for the Russell 1000, QP-OLS achieves a cross-sectional R^2 of 94% using 10 days of 10 second data. We expect that our results with 10 days of data are similar to what could be achieved using 1 day of 1 second data, assuming that intra-day variation in the mid-prices is comparable to the 10-day sample.

Dow 30	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	1.000	0.000	1.000	0.000	0.923	0.022	0.969	0.016
Equal	1.000	0.000	1.000	0.000	0.896	0.028	0.951	0.025

Naq 100	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	1.000	0.000	1.000	0.000	0.628	0.058	0.801	0.068
Equal	1.000	0.000	1.000	0.000	0.555	0.065	0.706	0.077

SP 500	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	1.000	0.000	0.999	0.000	0.041	0.015	0.107	0.037
Equal	1.000	0.000	0.999	0.000	0.033	0.013	0.085	0.026

Rus 1000	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	0.996	0.010	0.985	0.020	0.009	0.006	0.040	0.018
Equal	0.996	0.007	0.983	0.015	0.007	0.004	0.034	0.013

Portfolio Size: 20% N Portfolio: 100 Date of market prices: 08-01-2014

Table 1: For each stock universe, shows the R^2 of the true weights vs. those estimated from OLS and LASSO regressions of 10 second VIIV returns on stock returns for 100 random portfolios. 20% of each universe is randomly selected and given either equal weights ("Equal") or random positive weights ("Var Pos"). The first two columns show results using the exact VIIV, the last two for the rounded VIIV.

SP 500	OLS		Lasso		Rounded OLS		Rounded Lasso	
Var Pos	Mean	Std	Mean	Std	Mean	Std	Mean	Std
75 pct	1.000	0.000	0.993	0.001	0.013	0.007	0.079	0.019
50 pct	1.000	0.000	0.997	0.001	0.017	0.011	0.070	0.023
25 pct	1.000	0.000	0.999	0.000	0.034	0.015	0.091	0.033

SP 500	OLS		Lasso		Rounded OLS		Rounded Lasso	
Equal	Mean	Std	Mean	Std	Mean	Std	Mean	Std
75 pct	1.000	0.000	0.989	0.001	0.008	0.004	0.086	0.015
50 pct	1.000	0.000	0.996	0.001	0.015	0.007	0.077	0.014
25 pct	1.000	0.000	0.999	0.000	0.027	0.013	0.075	0.022

N Portfolio : 100 Date of market prices: 08-01-2014

Table 2: Using the S&P 500 as a stock universe, shows the R^2 of the true weights vs. those estimated from OLS and LASSO regressions of 10 second VIIV returns on stock returns for 100 random portfolios. The first column gives the percentage of the universe randomly selected to have non-zero weights, and the two sub-tables repeat this for both equal and variable positive weights. The first two columns show results using the exact VIIV, the last two for the rounded VIIV.

Dow 30	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	0.699	0.000	0.905	0.059	0.659	0.063	0.714	0.079
Equal	0.693	0.000	0.905	0.062	0.693	0.000	0.754	0.042

Naq 100	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	0.699	0.000	0.804	0.028	0.561	0.058	0.621	0.073
Equal	0.693	0.000	0.799	0.025	0.635	0.041	0.684	0.062

SP 500	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	0.696	0.003	0.785	0.015	0.167	0.041	0.194	0.050
Fixed	0.692	0.000	0.790	0.010	0.164	0.044	0.178	0.048

Rus 1000	OLS		Lasso		Rounded OLS		Rounded Lasso	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	0.694	0.006	0.768	0.023	0.092	0.031	0.106	0.038
Equal	0.693	0.001	0.787	0.008	0.087	0.029	0.105	0.034

Portfolio Size : 20% N Portfolio : 100 Date of market prices: 08-01-2014

Table 3: For each stock universe, shows the Spearman rank correlation of the true weights vs. those estimated from OLS and LASSO regressions of 10 second VIIV returns on stock returns for 100 random portfolios. 20% of each universe is randomly selected and given either equal weights ("Equal") or random positive weights ("Var Pos"). The first two columns show results using the exact VIIV, the last two for the rounded VIIV.

SP 500	OLS		Lasso		Rounded OLS		Rounded Lasso	
Var Pos	Mean	Std	Mean	Std	Mean	Std	Mean	Std
75 pct	0.991	0.001	0.980	0.002	0.079	0.045	0.074	0.047
50 pct	0.934	0.002	0.938	0.005	0.095	0.047	0.093	0.050
25 pct	0.758	0.003	0.825	0.010	0.151	0.044	0.165	0.049

SP 500	OLS		Lasso		Rounded OLS		Rounded Lasso	
Equal	Mean	Std	Mean	Std	Mean	Std	Mean	Std
75 pct	0.749	0.000	0.748	0.001	0.043	0.041	0.039	0.039
50 pct	0.866	0.000	0.880	0.002	0.092	0.046	0.089	0.042
25 pct	0.749	0.000	0.822	0.008	0.142	0.044	0.148	0.046

N Portfolio : 100 Date of market prices: 08-01-2014

Table 4: Using the S&P 500 as a stock universe, shows the Spearman rank correlation of the true weights vs. those estimated from OLS and LASSO regressions of 10 second VIIV returns on stock returns for 100 random portfolios. The first column gives the percentage of the universe randomly selected to have non-zero weights, and the two sub-tables repeat this for both equal and variable positive weights. The first two columns show results using the exact VIIV, the last two for the rounded VIIV.

Dow 30	Var Pos Lasso		Var Pos Rounded Lasso		Equal Lasso		Equal Rounded Lasso	
	False Positive	Matching	False Positive	Matching	False Positive	Matching	False Positive	Matching
Estimate Mean	2.07	6.00	8.48	5.72	1.91	6.00	9.05	6.00
Std	1.57	0.00	3.34	0.53	1.65	0.00	3.98	0.00
Real		6.00		6.00		6.00		6.00

Naq 100	Var Pos Lasso		Var Pos Rounded Lasso		Equal Lasso		Equal Rounded Lasso	
	False Positive	Matching	False Positive	Matching	False Positive	Matching	False Positive	Matching
Estimate Mean	16.66	18.95	20.68	15.66	16.75	19.00	23.25	17.77
Std	4.06	0.22	5.69	1.62	3.68	0.00	5.92	1.14
Real		19.00		19.00		19.00		19.00

SP500	Var Pos Lasso		Var Pos Rounded Lasso		Equal Lasso		Equal Rounded Lasso	
	False Positive	Matching	False Positive	Matching	False Positive	Matching	False Positive	Matching
Estimate Mean	93.20	95.11	55.16	28.67	89.93	96.00	58.31	29.33
Std	9.82	1.01	8.15	4.02	8.65	0.00	7.78	3.94
Real		96.00		96.00		96.00		96.00

Rus 1000	Var Pos Lasso		Var Pos Rounded Lasso		Equal Lasso		Equal Rounded Lasso	
	False Positive	Matching	False Positive	Matching	False Positive	Matching	False Positive	Matching
Estimate Mean	180.92	189.46	75.14	31.63	184.06	196.29	74.33	32.63
Std	12.85	5.74	12.88	5.92	11.57	0.87	13.53	5.99
Real		197.00		197.00		197.00		197.00
N Portfolio : 100		Date of market prices: 08-01-2014						

Table 5: For each stock universe, shows statistics for the average number of matching and false-positive stocks in estimated portfolios using the LASSO algorithm. "Real" denotes the true number of stocks in the portfolio. 100 regressions of 10 second VIIV returns on stock returns are run for both equal weights and random positive weights on 20% of the S&P 500.20%, using both exact and rounded VIIVs.

SP 500	OLS		Rounded OLS	
	Mean	Std	Mean	Std
Day 1	1.000	0.000	0.070	0.018
Day 2	1.000	0.000	0.156	0.019
Day 3	1.000	0.000	0.295	0.032
Day 4	1.000	0.000	0.392	0.037
Day 5	1.000	0.000	0.476	0.035
Day 6	1.000	0.000	0.538	0.047
Day 7	1.000	0.000	0.576	0.046
- - -	- - -	- - -	- - -	- - -
Day 44	1.000	0.000	0.954	0.005
Day 45	1.000	0.000	0.955	0.005
Day 46	1.000	0.000	0.958	0.005
Day 47	1.000	0.000	0.960	0.005
Day 48	1.000	0.000	0.962	0.005
Day 49	1.000	0.000	0.965	0.005
Day 50	1.000	0.000	0.967	0.005
Date of market prices: 08-01-2014			N Portfolio : 10	

Table 6: Shows the cumulative cross-sectional R^2 of true portfolio weights vs. those estimated from OLS regressions of VIIV *levels* on correctly scaled stock prices (as in Section 1) over multiple days of 10 second data. Results are averaged across 10 random portfolios with variable positive weights made up of 20% of randomly selected S&P 500 stocks.

Dow 30	OLS		Quad Prog		Rounded OLS		Rounded Quad Prog	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
VarPos	1.000	0.000	1.000	0.000	0.994	0.003	1.000	0.000
Equal	1.000	0.000	1.000	0.000	0.992	0.003	1.000	0.000

Naq 100	OLS		Quad Prog		Rounded OLS		Rounded Quad Prog	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
VarPos	1.000	0.000	1.000	0.000	0.888	0.026	0.997	0.002
Equal	1.000	0.000	1.000	0.000	0.856	0.027	0.995	0.002

SP 500	OLS		Quad Prog		Rounded OLS		Rounded Quad Prog	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
VarPos	1.000	0.000	1.000	0.000	0.063	0.018	0.471	0.073
Equal	1.000	0.000	1.000	0.000	0.052	0.022	0.327	0.047

Rus 1000	OLS		Quad Prog		Rounded OLS		Rounded Quad Prog	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Var Pos	1.000	0.000	1.000	0.000	0.012	0.006	0.155	0.029
Equal	1.000	0.000	1.000	0.000	0.009	0.005	0.116	0.024

Portfolio Size: 20% N Portfolio: 100 Date of market prices: 08-01-2014

Table 7: For each stock universe, shows the R^2 of the true weights vs. those estimated from OLS and constrained quadratic programming regressions of 10 second VIIV *levels* on correctly scaled stock prices for 100 random portfolios. 20% of each universe is randomly selected and given either equal weights ("Equal") or random positive weights ("Var Pos"). The first two columns show results using the exact VIIV, the last two for the rounded VIIV.

SP500	OLS		Quad Prog		Rounded OLS		Rounded Quad Prog	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Day 1	1.000	0.000	1.000	0.000	0.063	0.018	0.471	0.073
Day 2	1.000	0.000	1.000	0.000	0.153	0.026	0.776	0.048
Day 3	1.000	0.000	1.000	0.000	0.281	0.032	0.911	0.023
Day 4	1.000	0.000	1.000	0.000	0.383	0.032	0.959	0.011
Day 5	1.000	0.000	1.000	0.000	0.471	0.034	0.977	0.007
Day 6	1.000	0.000	1.000	0.000	0.532	0.035	0.985	0.004
Day 7	1.000	0.000	1.000	0.000	0.576	0.035	0.989	0.003
Day 8	1.000	0.000	1.000	0.000	0.616	0.030	0.992	0.002
Day 9	1.000	0.000	1.000	0.000	0.643	0.027	0.994	0.002
Day 10	1.000	0.000	1.000	0.000	0.656	0.029	0.996	0.001

Date of market prices: 08-01-2014 N Portfolio : 100

Table 8: Shows the cumulative cross-sectional R^2 of true portfolio weights vs. those estimated from constrained quadratic programming regressions of VIIV *levels* on correctly scaled stock prices (as in Section 1) over multiple days of 10 second data. Results are averaged across 100 random portfolios with variable positive weights made up of 20% of randomly selected S&P 500 stocks.

Rus 1000	OLS		Quad Prog		Rounded OLS		Rounded Quad Prog	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std
Day 1	1.000	0.000	1.000	0.000	0.012	0.006	0.155	0.029
Day 2	1.000	0.000	1.000	0.000	0.031	0.012	0.334	0.049
Day 3	1.000	0.000	1.000	0.000	0.060	0.016	0.540	0.049
Day 4	1.000	0.000	1.000	0.000	0.090	0.020	0.683	0.043
Day 5	1.000	0.000	1.000	0.000	0.122	0.023	0.788	0.033
Day 6	1.000	0.000	1.000	0.000	0.156	0.024	0.849	0.025
Day 7	1.000	0.000	1.000	0.000	0.176	0.028	0.885	0.022
Day 8	1.000	0.000	1.000	0.000	0.203	0.027	0.913	0.015
Day 9	1.000	0.000	1.000	0.000	0.221	0.027	0.931	0.012
Day 10	1.000	0.000	1.000	0.000	0.237	0.028	0.943	0.011

Date of market prices: 08-01-2014 N Portfolio : 100

Table 9: Shows the cumulative cross-sectional R^2 of true portfolio weights vs. those estimated from constrained quadratic programming regressions of VIIV *levels* on correctly scaled stock prices (as in Section 1) over multiple days of 10 second data. Results are averaged across 100 random portfolios with variable positive weights made up of 20% of randomly selected Russell 1000 stocks.