Distilled Impact

Allison Bishop[*]

Introduction

For trading in US equities, measuring slippage is important and also hard. There is little agreement among market practitioners on how best to do so, and the most common approaches have serious limitations. Slippage vs. arrival, for example, compares the volume-weighted average price achieved by a trader to the price of the security at the moment the trading began. This is an appealing metric because it compares something influenced by the trader's actions (the achieved price) to something not influenced by the trader's actions (the initial price). One would hope that this comparison would yield valuable insight into the efficacy of the trader's behavior. However, there are significant external factors that can affect the achieved price but not the arrival price, such as trading activity in other stocks or general market trends over the course of the trading activity period. We may expect some external forces to have effects that "average" to 0 with enough sample size, but how much sample size is enough? And what about general trends like long bull or bear markets that don't average out to 0? Often we may not have enough samples of trading activity to overcome the considerable noise in slippage vs. arrival and draw meaningful conclusions.

To circumvent this difficulty, many consider slippage vs. VWAP, which instead compares the volume-weighted average price achieved by a trader to the volume-weighted average price achieved by *all* trades in that same security across the same time period (or roughly the same time period). The intuition here is that confounding external forces should affect both numbers similarly, and hence should wash out of the comparison. This greatly reduces the noise in the measurement. But it introduces a circularity problem: we are comparing something influenced by the trader's actions to something also influenced by the trader's actions, though perhaps to a lesser extent. This creates a blind spot in which poor trade performance can hide.

The market is a complex beast, and it is challenging to separate one's own activity from what would have otherwise happened. Faced with these frustrating circumstances, many brokers give up on really understanding and improving their performance. Instead, they perform statistically flimsy transaction cost analyses to make themselves and their clients look good and to check the regulatory box of achieving "best execution." This is a much less challenging task, as there are many reasonable ways to slice the noisy data, and some of those ways are bound to yield "statistically significant" results.

The very notion of "statistical significance" was defined in a world before big data. Like *way* before. Like in the "100 people is a big data set" era. It was also an era before hordes of data science boot camp graduates armed with weapons-grade AWS accounts descended upon a sleepy, unprepared statistical theory. Today, we can run hundreds if not thousands of analyses with ease. With a healthy amount of noise in the data set, that means we can basically produce any result we want by trying all somewhat reasonable variations, as long as we only have to produce it once. And we don't even have to be deliberately cheating. We might simply be running

[*] Proof Trading, allison@prooftrading.com

analyses continually until something pops up as “significant.” If our threshold for “significance” is a 1-in-100 chance, say, and we run 100 tests, we should expect to find something “significant” by random chance alone. And the usual “two standard deviations” threshold is even worse - that’s a 1 in 20 chance!

Does this mean that measuring and improving broker performance is a hopeless task? Doomed to drown in a sea of noise and opaque, sloppy practices? Hopefully not. We think this defeatist attitude is premature. Noise can be combatted more effectively on two fronts: 1. better choices of metrics and 2. transparency and thorough documentation of context for all performed tests.

For metric design, there is a continuum of options between the noise-choked slippage vs. arrival and the illusory slippage vs. VWAP. Many of the external factors that slippage vs. VWAP seek to remove leave their trace in other places beyond the trading of that particular security. There is a wealth of data from that same time period in *other* securities that can give us clues to the external factors. Mining this data might help us avoid the full circularity of slippage vs. VWAP while still allowing us to reduce the noise in our measurements compared to slippage vs. arrival.

Our goal here is to define a new metric, distilled impact, for measuring the quality of execution for trading activity in retrospect. Our metric will use contemporaneous data from other symbols to inform our estimation of what the outcome might have been without our activity, ideally allowing us to form a less noisy and less circular baseline for comparison.

Our design of this metric is an ongoing research effort, and what we present here are our initial results. We’ll first present a summary of our work so far, but then we’ll also detail all of the discarded choices and tests we performed along the way. This includes tests that failed to produce the results we expected, and tests that reveal poor choices we made initially and later corrected. This tends to make our research reports longer than they would be if we simply described our final results. But hopefully it provides important context and it’s a habit we want to entrench as a mechanism for holding ourselves accountable. As a bonus, it’s ultimately more efficient: if we document our failures as well as our successes, we can continue to learn and build institutional knowledge from all parts of our research process, and not just the outwardly successful bits. “But I just want to see the statistically significant stuff!” some might say. “I trust you.” Sigh. There are some people you should never trust: real-estate brokers, lawyers, and statisticians.

Summary

We find that a rather basic approach significantly reduces the level of noise in price movements across the market. We use a short list of highly liquid ETFs as proxies for some price movements that are market-wide, sector-wide, or factor-wide. For each symbol, we compare the relative price movements in that symbol to the relative price movements in our proxy ETFs over many different data points, and we fit a linear combination of the ETFs that best correlates with price movements in that symbol. On fresh data, we then “distill” the relative price movements in each symbol by subtracting the customized linear combinations of the proxy ETFs’ relative price movements in the same time periods. What we are left with is “noise” in the movements of each symbol that is lower in magnitude and variance than it was before this correction for larger forces that operate across symbols.

This approach (once fit on a sufficiently large data set), appears quite robust to choices of different timescales, and reliably outperforms a more basic adjustment based on SPY alone. Nonetheless, we expect there is much room for improvement, and we view this as a preliminary version of our distillation method. We plan to iteratively improve this over time, ultimately removing as much noise as possible, further isolating the idiosyncracies of price movements in individual symbols and allowing us to reveal finer evidence of impact of trading dynamics within a single symbol.

Our starting point: A basic model of stock price movement

We will start by developing a basic model of stock price movement. There are many challenges inherent in formulating models of stock price movement. Even initial choices of units, like using a number of shares vs. notional value, or using absolute price differences vs. basis points, can have a chaotic effect on model outcomes if we are not careful. For this reason, we will try to stick closely to three high level principles:

1. Keep the model as simple as possible.
2. Keep the model as broad as possible.
3. Make assumptions as explicit and minimal as possible.

Essentially, we want to maximize the amount of data we can use to fit and test our model, and we want to make sure our design is robust to natural changes in market behavior over time, as well as mild violations of our underlying assumptions. Complicated models have a higher chance of being over-fit and fragile as trends evolve, and can sometimes deliver drastic differences in results from only minor differences in input.

These high level principles guide our initial choice of units: we want to begin with units that are well-defined over *all* stocks, and can be compared apples-to-apples across different kinds of stocks, regardless of price or level of activity. For this reason, we will look at "movement" in the prices of stocks in terms of basis points, normalizing by the prices so that we can meaningfully aggregate movements over stocks that individually have very different prices.

If we want to model the movement in the price of a stock S over a time period t as a function of various forces, we can start with some simple hypotheses as to what those forces might be. Here is an initial list:

1. significant actions taken by someone quoting/trading in stock S over the time period t
2. the general direction of market as a whole over the time period t
3. the aggregate direction of common sectors and factors over the time period t, to the extent they are correlated with S

In keeping with the principle of starting simple until we see reason to do otherwise, we'll formulate a simple additive model of these forces:

$$M(S,t) = A(S,t) + O(t) + C(S,t) + N(S,t),$$

where $M(S,t)$ denotes the relative change in the price of stock S from the beginning to the end of time period t, $A(S,t)$ denotes the impact of significant actions taken in the quoting/trading of stock S during time period t, $O(t)$ denotes the overall movement of the market during time period t, $C(S,t)$ denotes the combined movement of sectors and factors over time period t weighted by their correlation with S, and $N(S,t)$ is a "noise" or error term that catches the differences between the sum of these forces and the actual relative movement in the price of S.

Even this basic model sparks some critical questions: what counts as a "significant action"? Won't there be some overlap between movements in sectors/factors and movements in the overall market? Why should we allow symbol-specific weighting inside the formulation of $C(S,t)$ but not allow a weighting on $O(t)$? How will $O(t)$ be measured? What sectors/factors contribute to C and how will their weights be measured?

Naturally, as we begin to address these questions, we begin to make choices of specifics that we may need to revisit later in the case that they prove to be sub-optimal. But we cannot avoid making them if we are to arrive at something concrete. We'll start by using movement in SPY as a proxy for overall market movement, and we'll use a set of other ETFs as proxies for sector/factor movements. For now, we'll let the set of variables $\{c_i(t)\}$ denote the relative movements in these ETFs (including SPY) over the time period t, and we'll let $\{w_i(S)\}$ denote weights that depend upon the symbol S. Our reformulated model is:

$$M(S,t) = A(S,t) + \sum_i w_i(S)c_i(t) + N(S,t). \tag{1}$$

Notice that we have now allowed the "overall market" proxy of SPY to have a weighted coefficient. The assumption here (which we always endeavor to make explicit!) is that these ETFs are sufficient proxies for the kind of sector/factor effects that may contribute to driving price movements in the symbol S. There is also a fundamental assumption that the total combined effect of all of these forces is a linear function of the individual parts.

To keep ourselves grounded, we'll be comparing this model to the even simpler version:

$$M(S,t) = A(S,t) + O(t) + N(S,t), \tag{2}$$

where $O(t)$ denotes the relative movement in SPY over time period t.

Now, since we have allowed $N(S,t)$ to be an arbitrary term, neither model can be "wrong." For any values of $M(S,t)$, $A(S,t)$ and $\{c_i(t)\}$, there exists a unique value of $N(S,t)$ that makes equation (1) hold. Similarly, for any values of $M(S,t)$, $A(S,t)$ and $O(t)$, there exists a unique value of $N(S,t)$ that makes equation (2) hold. So which value of $N(S,t)$ is "right"? What does it mean to compare these models?

For either model to be useful in any way, we would want N to have some non-trivial features:

- We want the typical absolute value of $N(S,t)$ to be as small as possible.
- We want the expectation of $N(S,t)$ to be as close to 0 as possible.

These features seem intuitive: if N is intended to be an "error" term, it should be small and mean 0, otherwise something structured and significant is going into N. If that is happening, we should be finding new terms to add to the model to capture this meaning, rather than throwing it into the error basket. However, there are some subtleties lurking in the words "typical" and "expectation." Both of these words implicitly require us to define a *distribution* over symbols and time periods. Our definition of this distribution does not change the general form of our model, but it may change the *meaningfulness* of our model. Some definitions may result in $N(S,t)$ solidly satisfying the above criteria, and others may result in these criteria being violated to a prohibitive extent.

That said, we are not interested in defining a distribution *for the purpose of making $N(S,t)$ amenable*. We are interested in defining a distribution that is most aligned with our trading behavior and goals, and then finding ways to design $\{w_i c_i\}$ in equation (1) to make $N(S,t)$ and small and 0-centered as possible. With this in mind, we'll start with a time horizon of 1 day and a uniform distribution over symbols. Both of these choices we will soon revisit.

What we can see and what we wish we could see

The relative movement in the price of a stock S over a time period t is something we can measure from trade and quote (TAQ) data. Already there are several detail choices we need to make: do we look at quote prices or trade prices? At what exact moment do we sample the "price"? Hopefully reasonable variations of these choices do not make much of an impact on the ultimate outcomes, which is something we can try to confirm experimentally later. For now, for t equal to a particular day, we'll let $F(S,t)$ denote the price of the first trade that occurs in symbol S after 10:00 am on that day, and we'll let $L(S,t)$ denote the price of the last trade that occurs in symbol S before 3:30 pm. We've excluded the time from 9:30 am to 10:00 am and from 3:30 pm to 4 pm in order to avoid idiosyncracies of auctions and their effects. We'll define:

$$M(S,t) := (L(S,t) - F(S,t))/F(S,t). \tag{3}$$

It is worth pausing here to note our assumption that this approximately represents "the relative movement in the price of stock S over the course of the day t." Seems reasonable on its face, but some things could go wrong. What if there are no trades in this time period? What if there is only one trade in this time period? What if there are no trades near 10 am or no trades near 3:30 pm? In those cases, does this calculation still capture the intended spirit of $M(S,t)$? Probably not. But for now we will accept this, as we do not expect such cases to have a large effect on our overall computation. It is certainly something to keep in mind though.

Using the definition in (3), $M(S,t)$ is something we can calculate from historical TAQ data for any symbol/day pair. But is probably not that meaningful to us in isolation. If we are working from the perspective of an agency broker, trading orders in given symbols over certain days, we are more interested in the *impact* of $M(S,t)$ on our clients' bottom line for the symbols/days that we are trading. To capture this, we need to introduce a few more variables. We'll let σ denote an "order" consisting of a symbol S, a day t, and a side $\Gamma \in \{+1, -1\}$, where $+1$ corresponds to a buy order and -1 corresponds to a sell order. We'll let NV denote the notional value traded that day in that symbol as part of the execution of the order σ. The total impact to the client's bottom line, relative to slippage vs. arrival, can then be expressed as:

$$\sum_{\sigma} \Gamma * M(S,t) * NV, \tag{4}$$

where Γ, S, t, and NV vary appropriately as functions of the order σ. Note that because of our chosen conventions for the signs Γ, we should want to *minimize* the value in (4). This value is something we can compute experimentally for our trading behavior. But attempting to minimize it directly (e.g. A-B testing different versions of our algo and comparing them based on the value of (4)) would be a very noisy and hence error-prone process. Later, we will give some experimental evidence of just *how noisy* this quantity might be. But for now, let's try to further refine our target quantity to something that might be less noisy, but still captures our intent to minimize the toll of transactions costs to our clients.

If we substitute in our model in (1) for $M(S,t)$, we can rewrite (4) as:

$$\sum_{\sigma} \Gamma * (A(S,t) + \sum_{i} w_i(S)c_i(t) + N(S,t)) * NV \tag{5}$$

We can write this equivalently as:

$$\sum_{\sigma} \Gamma * A(S,t) * NV + \sum_{\sigma} \Gamma * NV \left(\sum_{i} w_i(S)c_i(t) \right) + \sum_{\sigma} \Gamma * N(S,t) * NV$$

If our model functions as we intend it to, and we are trading relatively light amounts that are unlikely to move the market as a whole or move an entire sector/factor significantly, only the first term of these three terms is meaningfully responsive to our actions. So what we'd really like to directly minimize is:

$$\sum_{\sigma} \Gamma * A(S,t) * NV.$$

Minimizing this would likely mean trying to make it close to 0, as if we do act in a way the moves the stock price of S, we are likely to push it up when we are buying, and push it down when we are selling. Hence we suspect that this term will generally be positive, and our goal is to make it as close to 0 as possible.

But this term is not something we can measure directly, as we don't actually know how to break $M(S,t)$ into these component parts on an order by order basis. A natural thing to do, though, is to try to get as close to this term as possible by subtracting the second term in the sum above, which we can calculate:

$$\sum_{\sigma} \Gamma * \left(M(S,t) - \sum_{i} w_i(S) c_i(t) \right) * NV = \sum_{\sigma} \Gamma * A(S,t) * NV + \sum_{\sigma} \Gamma * N(S,t) * NV \quad (6)$$

This will be a better proxy for $\sum_{\sigma} \Gamma * A(S,t) * NV$ if we design the subtracted terms $\sum_{i} w_i(S) c_i(t)$ to make $\sum_{\sigma} \Gamma * N(S,t) * NV$ as small as possible. But of course, we also cannot directly measure $\sum_{\sigma} \Gamma * N(S,t) * NV$.

So what are we to do? Well, one approach is to re-conceptualize $A(S,t)$ as only meaning the impact of *our trading* in symbol S over time period t, and to lump the impact of anyone else's trading into the definition of $N(S,t)$. This feels like a reasonable thing to do, since our impact is what we really want to minimize, and so the spirit of minimizing $\sum_{\sigma} \Gamma * A(S,t) * NV$ holds under this interpretation. Also, our intuitive notion of trading impact is something that typically reverts, so we might guess that the overall expected value of other people's trading impact should be 0 over time. This is potentially a problematic assumption, so we should keep this closely in mind going forward.

Noting these caveats and moving boldly forward nonetheless, we might assert that in historical TAQ data where we were *not trading at all*, $A(S,t)$ is 0, and we can directly measure

$$\sum_{\sigma} \Gamma * N(S,t) * NV = \sum_{\sigma} \Gamma * \left(M(S,t) - \sum_{i} w_i(S) c_i(t) \right) * NV.$$

And hence, our goal is to design values of $w_i(S)$ and $c_i(t)$ such that the quantity

$$\sum_{\sigma} \Gamma * \left(M(S,t) - \sum_{i} w_i(S) c_i(t) \right) * NV \quad (7)$$

has expectation close to 0 and has a small a variance as possible, for distributions of σ that are "like" the orders we expect to receive from clients.

Since these orders are still hypothetical, we have to make sense of the term NV without reference to an actual volume traded in the execution of an order. Instead, we'll just use the total notional value traded in that symbol over that time period. You might object that surely this is too much: e.g. perhaps 1% of this, or some other constant fraction, is a more reasonable approximation to what would really correspond to a single parent order on that day. However, this doesn't matter: any fixed constant multiplied into this term for every order would pull out in front of the sum and have no effect on our task of minimization.

As a default for our distribution of symbol/day pairs, we will simply use a uniform distribution. This means each symbol is considered equally likely to be traded, and trading is equally likely to happen on each day. This may seem a little odd since some symbols represent much more overall trading activity than others. However, recall that we are supposing that an order in an symbol/day pair participates *proportionally to the notional value traded for that symbol over that day.* This makes a uniform distribution over the symbol/day pairs seem more reasonable. For now, our definition of the universe of "symbols" is all symbols that appear in our TAQ database (supplied by our historical market data provider, OneTick). This as well as the uniform distribution on symbol/day pairs should be heavily scrutinized and we'll return to this later.

We will also assume a uniform distribution on Γ (i.e. probability $\frac{1}{2}$ on $+1$ and probability $\frac{1}{2}$ on -1 for now, which, if this is chosen independently of everything else, assures that our random variable in (7) has expectation equal to 0. Its variance is then equal to:

$$\sum_{\sigma} \left(M(S,t) - \sum_{i} w_i(S) c_i(t) \right)^2 (NV)^2 \tag{8}$$

where the sum is over all symbol/day pairs in whatever data set we use for model testing.

An initial design and experimental results

We started with a short list of proxy ETFs for the c_i's: SPY, XLE, XLF, XLK, XLV, IWC, and SPVU. SPY was chosen as an overall market proxy, XLE, XLF, XLK, and XLV were chosen as sector proxies, and IWC and SPVU were chosen as factor proxies (according to https://www.portfoliovisualizer.com/etf-and-mutual-fund-factor-regressions, they are well-correlated with respective factors of the 3-factor Fama-French model).

We considered two approaches for defining $w_i(S)$. As a first approach, we took each symbol S and computed

$$\sum_t (M(S,t) - c_i(t))^2 (NV)^2$$

for each value of i, summing over all trading days t in the year 2018. We compared all these of these values, as well as the value of

$$\sum_t (M(S,t))^2 (NV)^2.$$

If the raw value of $\sum_t (M(S,t))^2(NV)^2$ was the smallest, we set $w_i(S) = 0$ for all i. If some other value was the smallest, we set $c_i = 1$ for that value of i and $c_j = 0$ for all $j \neq 0$. We refer to this as the "single proxy" method.

Our second approach fit a linear regression to the data from 2018 to find the least squares best fit for $M(S,t)$ as a linear combination of $c_i(t)$ as well as a constant term. Technically, this adds $c_0 = 1$ as a constant function in order to be consistent with the notation above. Notably, we did not yet weight the regression according to the notional value on each day, and instead treated all days in the training sample equally. (We will come back and fix this later, and the impact will be mild.) We naturally assigned the coefficients w_i to be the coefficients obtained by the linear regression. We refer to this as the "linear regression" method.

We then calculated the values of (8) for the coefficients obtained by each method, summing over all the trading days and symbols in 2019. Note that this test data is *disjoint* from the training data from 2018. (There is some subtlety to this overall summation, as there may be symbols appearing in the test data that were not in the training data. Those symbols were removed. We only calculated the values for symbols that appeared in our training set. There is a further subtlety due to the design of OneTick - when we ask it to automatically generate a list of all the symbols occurring in the TAQ database, we have to specify a date. We believe that whatever date in 2018/2019 we specified then got used as the symbol list for all dates when we expanded the date range to all of 2018 or 2019 for subsequent queries. We suspect that symbols that were not listed on the specified symbol generation date were dropped from the overall sum. We did not chase down this detail, as we doubt de-listed or newly listed symbols would have a big effect on our results in aggregate, but it's worth noting these kind of details just in case.)

For comparison purposes, we also computed the quantity in (8) over the same data set with all coefficients w_i set to 0 (we'll call this the "undistilled" method) and all coefficients set to 0 except a coefficient of 1 for SPY (we'll call this the "SPY only" method).

Here are the numbers obtained from OneTick (using TAQ data for 2019):

Single proxy	794.6834
Linear Regression	709.4585
SPY only	853.8029
Undistilled	1141.711

The units here have been adjusted by 10^{-16} to avoid the sums getting too large. (In other words, the raw values are these values times 10^{16}.)

We can see that the raw, undistilled metric that corresponds directly to slippage vs. arrival has the largest variance. In comparison, the SPY only method achieves a $(1141 - 853)/1141 =$ 25% reduction in variance, the single proxy method achieves a $(1141 - 795)/1141 = 30\%$ reduction in variance, and the linear regression method achieves a $(1141 - 709)/1141 = 38\%$ reduction in variance.

This is a proof of concept that shows distillation can be effective in reducing variance in our estimations of market impact. A few things to remember:

1. We made no effort to choose a comprehensive list of sector/factor ETFs. We threw in a small number just to see if it worked.

2. We did not yet weight the regression training data by notional value, which we would expect to perform slightly better.

3. We are currently including ETFs as part of our summation over all symbols. This likely makes our results look a little better than they would otherwise, as we suspect ETF price movements will be better captured by this style of distillation than the price movements of common stocks.

4. We should revisit our default distributions over symbols, time periods, and sides for orders if we have a better idea as to what will fit our clients' typical order flow.

5. We should come up with sanity checks for our assumptions wherever possible.

A first refinement

We went on to make some progress in addressing questions 1., 3., and 5. above, and naturally in the process we generated more questions.

For 3., we reran the numbers on the test data from 2019 and this time excluded ETPs. It did indeed reduce the level of improvement seen by comparing the undistilled number to the linear regression as we expected, but the remaining improvement was still quite meaningful. For 1., we threw in a fuller list of sector ETFs (one for each sector), and did not see a significant improvement as compared to the more limited selection above. This is likely an indicator that the amount of training data we are using is insufficient to support regression for that many variables on a per symbol basis. [Aside: you might wonder why we do not report here the exact results of these tests. The reason is we ran them, were unexcited by the results, forgot to save the numbers, and subsequently changed the code, making it pain to reproduce. This incident has sparked a larger thought process of how we should save and document all test results, even (or especially!) boring/disappointing ones, so stay tuned for a new and improved documentation process for our future research.]

A sign of trouble pops up when we make a reasonable modification to our methodology. We originally choose to measure: $M(S,t) := (L(S,t) - F(S,t))/F(S,t)$. This means our notion of "movement" is a stock price over a time interval is how much the last trade price in the interval deviates from the first trade price in percentage terms. It would seem equally reasonable to measure instead the relative deviation from the first trade price to the volume-weighted average price of trading over the interval. We would expect the magnitude of this measure to be a bit smaller on average, but qualitatively we'd expect it to behave similarly.

Well, if we switch to measuring $M(S,t)$ this way and adjust the $c_i(t)$ terms analogously while keeping the same weights w_i that we trained previously, we can calculate the quantity in 8 again over our test data. If we compare it as before the analogous calculation with all $c_i = 0$ (undistilled) and with just $c_{SPY} = 1$ (SPY only), we get very disappointing results. The linear regression outperforms the raw undistilled version, but this time fails to outperform the SPY only version. Since putting all of the weight on SPY is an option our regression training has, this strongly indicates that at least one of three things is true: 1) the difference in method to define $M(S,t)$ causes considerably different behavior, 2) our training data and test data are considerably different in character, or 3) our data size is insufficient to support our model complexity, and we are suffering from over-fitting.

Since 1) would seem unlikely, we suspect the answer is some combination of 2) and 3). One sanity check to do here is to see how different the model performance looks on the training data vs. the testing data. Running our calculations on the training data (treating it as if it is the test data), we do indeed see markedly better improvement from distilled impact as relative to the undistilled data (compared to what we see on the testing data). This is evidence for effects 2) and 3) in combination, but doesn't really tease out to what extent each is happening.

So let's re-examine our choice to use 2018 data for training and 2019 data for testing. The total notional value traded in 2018 was considerably higher than the total notional value traded in 2019. Also, there were more days with atypically high notional value in 2018, and those high notional value days can be several times higher than typical days. This all suggests that systemic differences between our training and testing data sets might be significantly affecting our results, and also that weighting solely by total notional value across time scales like years might lead us place considerable weight on outliers.

It's not immediately clear that this is undesirable. After all, those unusually heavy notional value days might represent proportional trading gains or losses that we care about. But even if we *care* about trading during markedly atypical times, we can't really hope to model it very well. By definition of it being atypical, it is a small data set. Trying really hard to model a small data set well is exactly how we fall into traps like over-fitting.

So what can we do to decrease systemic differences between our training and testing data sets? One way to try to decrease systemic difference is to sample both training and testing data sets more uniformly over the two years of data we have. So instead of splitting by year, we'll alternate days: every other day will be considered training data, and the next day will be considered testing data. This gives us two roughly equal sized disjoint data sets that should be more similar in character.

Now that we believe our training and testing data sets are more reasonably uniform, what can we do to further decrease any effects of over-fitting? The most straightforward way to reduce over-fitting would be to increase our amount of data (relative to a fixed complexity of model). Getting data for more trading days is problematic though: it costs money in market data fees, it increases our computational needs, and it may not help all that much, as the older the data gets, the more likely it is to reflect differences in behavior over time that make it less useful for predicting current behavior. Hmmm... what to do. We'll do what computer scientists often do when faced with unfriendly constraints: we'll look for a way to cheat.

How to get more data without getting more data So far, we have been looking at each trading day as a single data point for each symbol. But if we believe there is meaningful correlation between the relative price movement in a given stock over a day and the relative price movement in appropriately chosen ETFs, it seems reasonable to believe that this correlation might be present at shorter time scales. If we divide the trading day from 10 am to 3 pm, for example, into 10 minute intervals, and treat each as a separate data point, we suddenly have 30 data points per symbol per day instead of just one.

Of course, this amplification of data isn't free. If it had no negative consequences, naturally we would continue doing it, chopping time into finer and finer pieces and making our data richer and richer. But there are several potential pitfalls we must keep in mind. The most obvious perhaps is the consumption of computational resources. More data means more memory is needed to store that data, and more time is needed to process that data and compute a regression. In our case, this is not yet our limiting factor. Another concern is the time scale of the correlation we are trying to model. If price movements in a given stock "tend to be" meaningfully correlated with price movements in SPY and other ETFs, over what time intervals does that correlation express itself? It might be correlated at the level of hours, but not at the level of minutes. Or it might be correlated at the level of minutes, but not at the level of seconds, etc. If we try to chop the data into finer time scales than the phenomena we are trying to model, the phenomena will disappear.

A third concern is more specific to our exact problem here. Our measurement of stock price "movement" for stocks and ETFs over a specified interval of time is a function of trades that happen in that time window. If these windows get too small relative to the frequency of trading activity in a given stock or ETF, these measurements will begin to lose their tether to our intended meanings. A time interval that has no or little trading in one of our chosen ETFs is particularly problematic, as we will struggle to define the $c_i(t)$ value that we want to include in our regression fitting. In fact, looking at the relatively light trading activity in $SPVU$, this is already a problem for 10 minute intervals. It seems the other ETFs we've chosen have sufficient liquidity to support 10 minute intervals.

And so we'll use this "larger" data set to fit our regression coefficients for the six remaining ETFs: SPY, XLE, XLF, XLK, XLV, and IWC. We'll do this on our new choice of training data set, which consists of data from every other day in 2018-2019. We'll also evaluate our results a little differently this time. Since variance of a quantity weighted by notional value will be heavily impacted by high notional value days, we're going to look instead at a different quantity, and first compute it individually for each day d. Instead of squaring, we'll take the absolute value of $M(S,t) - \sum_i w_i(S)c_i(t)$ for each 10 minute time interval t and each symbol S, and then we'll take the weighted average of these absolute values over the day, weighting each time interval, symbol pair by the notional value traded in that symbol over the time interval on that day, denoted by $NV(S,t)$:

$$\sum_{S,t\in d} NV(S,t) * \left|M(S,t) - \sum_i w_i(S)c_i(t)\right| / \left(\sum_S NV(S,d)\right) \tag{9}$$

Intuitively, this quantity should give us a sense of "large" the quantity $M(S,t) - \sum_i w_i(S)c_i(t)$ tends to be in magnitude on this particular day, weighting by notional value. We'll also want to combine these daily quantities into a single number for a quick comparison of our regression model to raw, undistilled data (just taking $|M(S,t)|$) as well as to a SPY only adjustment. For this, we'll take a raw average over days. So we are weighting *within* each day by notional value, but we are treating all days equally and not weighting by the relative notional value of days. This is intended to avoid undue influence of especially high notional value days.

We'll remove ETPs from our data set and compute solely over common stocks. If we stick with our definition of $M(S,t)$ that compares the last sale price over the interval t to the first sale price, we get the following averages of the daily numbers. Here "ABS-NV-DELTA" denotes the average over days of the notional value weighted average absolute value of the raw $M(S,t)$, "ABS-NV-SPY" denotes the analogous quantity for the absolute value of $M(S,t)$ minus the relative movement in SPY over the same period, and "ABS-NV-DI" denotes the analogous quantity for the absolute value of $M(S,t)$ minus the linear combination of the movements in our ETFs, with the coefficients produced by our regression.

	ABS-NV-DELTA	ABS-NV-SPY	ABS-NV-DI
training data	0.002515306	0.002258263	0.002033333
testing data	0.002512054	0.00225996	0.002082873

There are several encouraging things about these numbers. Firstly, the raw behavior of $M(S,t)$ as reflected in the "ABS-NV-DELTA" column and the SPY-adjusted behavior in "ABS-NV-SPY" is very similar for our training and testing data sets, suggesting that our data sets are reasonably similar. Also, adjusting by SPY seems to reduce the absolute value relative to the raw $M(S,t)$, and using our linear regression seems to reduce it further, as we would hope. The behavior of our regression is understandably a little better on the training data, but not so much better that it is worrisome. We may have found our way clear of over-fitting.

But before we declare victory, let's do another sanity check. Let's change our definition of $M(S,t)$ to compare the volume-weighted average price in the time interval to the first sale price. Our expectation is that this should make all the numbers somewhat smaller, but that the qualitative behavior should be the same. And indeed, we get:

	ABS-NV-DELTA	ABS-NV-SPY	ABS-NV-DI
training data	0.001604932	0.001443785	0.001326069
testing data	0.001603515	0.001446027	0.001352627

So while these reductions in the absolute values of the price movements are not too dramatic, they are significant, and now reliably behaving as we expect they should. This is a good sign.

But have we lost predictive power for longer time intervals? We might still worry, though, that while things look good for 10-minute time intervals, we may have lost predictive power on longer intervals that we still care about. To address this, we'll keep using our coefficients trained from 10-minute intervals, but we'll test their behavior on longer intervals. Here's what happens if we compute daily NV-weighted average of the same absolute values on hourly intervals instead of 10 minute intervals on our test data:

	ABS-NV-DELTA	ABS-NV-DI
testing data	0.005618829	0.004624747

And here's what happens if we look at daily intervals instead of hours:

	ABS-NV-DELTA	ABS-NV-DI
testing data	0.009482303	0.007881699

Again, this conforms roughly to our hopeful expectations. Our linear regression continues to significantly and consistently reduce the absolute values of price fluctuations at these larger time scales. Unsurprisingly, the fluctuations themselves tend to be larger on larger timescales to begin with.

A second refinement

There is still an issue we glossed over too quickly: the (il)liquidity of IWC. We said above the liquidity of our chosen ETFs (after removing SPVU) was sufficient to support 10-minute intervals, but we previously checked this in a limited sense. Previously, we removed from our training and testing data sets any 10-minute intervals where there was insufficient trading activity to compute a reasonable delta value for any one of our chosen ETFs (including IWC). More precisely, we tossed out any interval where there weren't at least 5 trades in each ETF. We then checked that the performance on the remaining data set was reasonable, indicating that we had sufficient sample size remaining to fit well-behaved regressions. However, we did not check at the time how many 10-minute intervals were being discarded.

Ultimately, we want our distilled impact metric to be broadly applicable as a tool in transaction cost analyses and algorithm design. Hence, we care not only about the model being reasonably well-behaved, but also about the model being well-defined in nearly all situations. Let's consider 10-minute intervals between 10 am and 3 pm on each trading day in 2018 for example. They are 30 such intervals each day, and approximately 250 trading days in the year, yielding roughly 7,500 such intervals. Let's see how many of these intervals contain at least 5 trades in SPY: it turns out to be 7,497 (so basically all of them). Similarly, 7,495 of these intervals contain at least 5 trades in XLE, and the same roughly holds for XLF, XLK, and XLV. For IWC, however, the situation is quite different. Only 4,786 of these intervals contain at least 5 trades in IWC, making it too illiquid to support a broadly applicable model.

As a result, we went back to replace IWC with another factor-based ETF, hoping to find a more liquid choice that would perform at least approximately as well. We decided to try VB, which contains at least 5 trades in 7,494 of the mid-day 10-minute intervals in 2018. We trained our model again, this time with SPY, XLE, XLF, XLK, XLV, and VB as the ETFs. We tested the new model on data from January through June of 2018 (using our original method of calculating delta by subtracting first prices from last prices and dividing by first prices):

	ABS-NV-DELTA	ABS-NV-SPY	ABS-NV-DI
testing data	0.00240159	0.002157249	0.002016799

We then performed the test with the alternate calculation for delta, using volume-weighted average price compared to first price:

	ABS-NV-DELTA	ABS-NV-SPY	ABS-NV-DI
testing data	0.001527076	0.001379834	0.001314862

[Aside: you might wonder why we went with a six month testing period here, rather than the full 2018-2019 data set we used previously. The reason is constraints on our computational resources in OneTick. Since we are now typically running other computations on our OneTick instance at the same time due to other ongoing research efforts, our memory and time resources are recently under strain. This was less true at the earlier time when we ran the full two year tests. Hence running for shorter test periods now is more attractive, and especially since the results are so in line with what we saw on the larger data set earlier, there is no compelling reason to continue testing on the full two year span.]

These results indicate that we have not really lost efficacy by replacing IWC with the more liquid VB. Hence, for now, our distillation method will use linear regressions trained per symbol on the set of $\{SPY, XLE, XLF, XLK, XLV, VB\}$ as proxies, and we will use 10-minute trading intervals for training purposes. These same coefficients appear still reasonable to use for evaluation at different timescales (e.g. 1 hour or 1 day intervals) and to use with different definitions of delta (e.g. last price vs. first or VWAP vs. first price).