Pretrade Analysis: A Delicate Exercise in Hubris and Humility

Allison Bishop[*]

1 Introduction

Pretrade analysis addresses the following scenario: we are considering trading a large amount of stock, and we want to predict what might happen if try to trade it over various timescales. What will happen if we try to complete the order within an hour? Within a day? Within 3 days? How much more should we expect to pay in total implicit and explicit costs if proceed very aggressively versus very passively? Trying to formulate answers to these questions before we start trading is typically called "pretrade analysis." Pretrade analysis is a challenging task because trading takes place in a complex interactive system. Since actions feed off each other as well as constantly shifting external conditions, it is impossible to assert with any certainty how the market will react to trading that has not yet occurred. Nonetheless, there is some method behind the madness. Underneath the chaotic veneer of the US equities market, there are intentional mechanisms at play, and these mechanisms can result in noticeable patterns. We will try to build a model of these patterns, informed by historical data. We will use this model to make predictions about what outcomes we might obtain when we take various trading actions.

At every step of our model design, we will maintain a brazen confidence that outcomes *can be* meaningfully predicted, as we could be paralyzed otherwise. But each time we take a step, we will look nervously behind us, concerned that we may have fallen off the meaningful path and desperate for signs of reassurance. This is how we will (hopefully) maintain the right balance of forward momentum and humble introspection that can lead us to meaningful model. It is a favorite saying in statistics that "all models are wrong, but some are useful" (attributed to George Box). This is profoundly true. And the path to a wrong but useful model is best pursued by those who hold both the wrongness and the usefulness close to the chest at all times.

2 Overview

The research we present here culminates in a model for how the distribution of likely prices changes as a function of how much directional volume we might insert into the market over a specified time scale. That's a mouthful, so let's break it down piece by piece.

At any given moment in time, we tend to think of the price of a particular stock as a number. Like "$50 per share." But that's too simplistic. Market participants consider multiple kinds of price information for a single stock at a given time. There are various quoted prices for buying/selling various amounts: e.g. "100 shares available that you can buy for $50.01 per share", "looking to buy 200 shares and willing to pay $49.99 per share," etc. And there are also recently completed transactions: "100 shares sold for $50.00 a share." Quotes and trades are continuously updated as activity happens across the fragmented market. For simplicity, we'll focus on the prices of completed trades (often referred to as "Last Sale" prices). In a given

[*] Proof Trading, allison@prooftrading.com

stock, the time series of transactions as they are reported form an ordered sequence of trade prices. At a given moment, we can define the “last sale price” as the price per share that was paid in the most recently reported transaction. This price is a single number at any particular moment in the past, but more generally can be viewed as a function of time.

Now, if we are looking ahead to a particular moment in the future, the most recent transaction price at the moment will also be a number, but it’s a number we can’t know yet. We can accommodate some of our uncertainty about the future by thinking of this future price not as a single number, but as a probability distribution. This should be a familiar mind set to anyone who plays games with cards or dice. Once a die is cast, the outcome is a single number between 1 and 6, but *before* the die is cast, we can think of the future value as having a $\frac{1}{6}$ chance of being a 1, a $\frac{1}{6}$ chance of being a 2, etc. This collection of probabilities (the amount of chance associated with each possible outcome) is called a probability distribution. This is a useful tool for reasoning about future events that have an element of randomness to them. A probability distribution can reflect whatever information we know so far. For example, if we are playing a game of poker, and some number of cards have already been revealed, we will know some cards are no longer in the deck, so we will treat them as having a 0 chance of being the next dealt card. Poker is an interesting example, as once the deck has been shuffled, the next card that will be dealt is already determined. There is no actual randomness left in the outcome! Nonetheless, we as the players do not yet *know* the outcome, so our imperfect knowledge can still be meaningfully described by a probability distribution.

As market participants, it makes sense for us to think of future prices as probability distributions. We will try to incorporate all of the knowledge we have into these distributions, but we cannot expect to have certainty about the future prices. This does have a consequence that is somewhat confusing: when we talk about past prices, we are talking about numbers. Familiar, comforting numbers of dollars and cents that are easy to wrap our heads around. But when we talk about future prices, we are talking about probability distributions on numbers. These distributions can be talked about with statements like “there is a 20% chance that the price will rise by at least 20 cents between now and the end of the day” or “there is a 75% chance that the price one hour from now will stay within 10 cents of the current price...” These kinds of statements are already highly technical, but even they leave out some very important caveats. With dice and cards, it’s clear where the underlying notion of “chance” comes from: it comes from the physical process of rolling the die or shuffling the cards. By making reasonable assumptions about these physical processes (specifically that each outcome of the die is equally likely or that shuffling is a truly random permutation of the cards), we can derive precise and defensible probabilities for each outcome, conditioned on whatever partial information we may have. With markets, the sources of “randomness” are much more opaque. As a result, many economists and analysts pretend that market outcomes are driven by invisible random processes that have very familiar and convenient structures: e.g. Gaussian processes, which distribute outcomes along classic bell-shaped curves. When such analysts say there is a particular “chance” of some event happening, they are implicitly saying: “assuming the market behaves like a Gaussian process with mean X and variance Y, the chance of ...”. Leaving off that technical preamble produces a much catchier sound bite, but the result is often misleading.

Naturally, people don’t just pick their assumed structures wholly out of thin air. They might assume a Gaussian process, but then find the values for the mean X and variance Y that are most consistent with past market data. We will try to assume as little as possible at first, and use data collection to build up a picture of the typical distribution of future prices under various circumstances. It is fairly straightforward to build our understanding of what “typical” market behavior is. We can take all of the prices we have observed over the recent past in our historical data set and use this empirical data to fill out a distribution. For example, if we want to know how often the last sale price increases by roughly 1% from the beginning to the end of a trading day, we could simply look at all of our observations, count how many times this happens, and divide by the total number of observations. This is probably a bit too simplistic: we can further refine things by dividing our observations into groups of similar stocks, and we may want to weight things by notional value instead of counting all observations equally. But the general principle is the same: we can suppose that what we see in the data represents "typical" behavior, and hence we can extract a reasonable model of typical behavior from our data.

It gets much trickier when we want to understand what a typical *response* to our trading behavior might be. Our historical transaction data specifies times, sizes, and prices of trades, but it does not identify market participants. A priori, we don't know when the trades we're looking at in the data set represent the actions of a large buyer or a large seller. Naturally, once we start live trading, we will know our own trades and we can try to look at metrics that capture market behavior in response to our trades and compare them to those metrics computed in other times, but this is kind of research is severely hindered by the small sample size of our own trades. And so we want to get what we can out of studying the vastly larger data set of historical prices across the market.

To address this challenge, we'll pair up historical samples of trading activity that seem to differ approximately by the kind of activity we are contemplating. We'll use these paired samples to build a distribution of what we expect to happen to prices when we perform a large trade. This immediately raises some questions: 1. how do we identify activity like "buying a large amount of stock over the course of a day" when individual trade data reports don't tell us the side of the trade? (i.e. whether the buyer or seller expressed more urgency). And 2. how do we combine these pairs of samples into a model of the distribution of price changes that might result from our trading? We will discuss these questions and several possible answers in detail below. For now, it suffices to think of each pair of samples as a (noisy) window into what might have been. Our hope is that collecting enough noisy data points yields an averaged view that is meaningful.

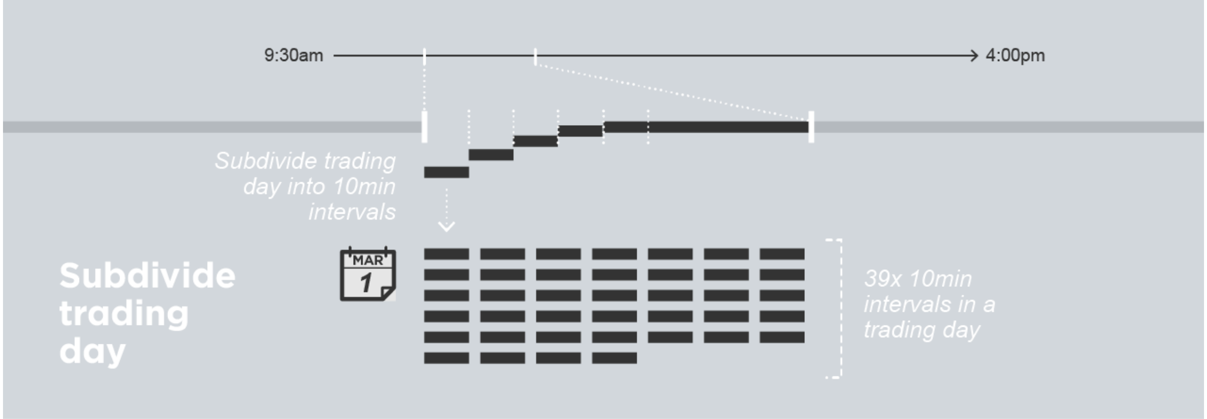
Ultimately, we'll arrive at a tool that allows us to ask questions of the form: if I were to buy X shares of symbol S over the course of one day, what's the probability distribution that expresses how the price of S might move as a result? Once we have a model that generates these probability distributions, we can extract information like the median expected price movement or the 25th percentile of expected price movements, etc. [Aside: it is common elsewhere for the mean and variance of distributions to be extracted and used to make decisions. In fact, we began by considering means/expectations in this work. We ultimately gravitated towards medians and percentiles instead, as means and variances can be heavily affected by outliers, while medians/percentiles are more robust.]

The rest of this paper chronicles the research that led to our model and accompanying pretrade analysis tool (http://pretrade.prooftrading.com) in its current form. We choose to explain our research process completely and chronologically from the beginning, so that we can explain why we took each step we took in proper context. This means many of the things we describe early on end up being changed by the time we arrive at the final model. It also means that this paper is much longer and much more winding than a direct, top-down description of our final model. For readers who only want to see a more concise explanation of our final model, you can find such an explanation in Proof's blog on medium (the relevant post is titled "To Serve Traders").

3 Getting down to details: defining units and market conditions

We'll start our research process by breaking the trading day into disjoint windows of time, and then we will summarize the trading activity and price change for each symbol in each window. The initial idea is to study how price distributions change as a function of trading activity. This can be a first step to understand how our impact on trading activity may translate to impact on price.

We'll start with each window being 10 minutes long, though we'll consider other window lengths later.


9:30am
4:00pm
Subdivide trading day into 10min intervals
Subdivide trading day
MAR
1
39x 10min intervals in a trading day

How we choose to summarize market activity in each time window is very important and will effect everything going forward. This is a key moment that we will revisit again and again to mine for future improvements to our model, but for now, we have to start somewhere. Once we fix a method of measuring and summarizing, we'll refer to the outputs of this process as market profiles. Along the way, we'll try to enumerate features of "good" market profiles that we can test for, and use these to see if we are following a reasonable path.

To construct our initial profiles on a per symbol basis, we'll take each trade and label it as below the prevailing midpoint price, on the midpoint price, or above the midpoint price. The intuition for this is that trades at prices below the prevailing midpoint signal that the seller was more urgent, and may exert more of a downward pressure on prices than trades at or above the midpoint. Conversely, trades at prices above the prevailing midpoint signal that the buyer was more urgent, and may exert a more upward pressure on prices. Once the trades are labelled in this way, we can sum up the volume in our time interval in each category. We'll call the sum of volume below the midpoint $DOWN_v$, the sum of volume at the midpoint $NEUTRAL_v$, and the sum of volume above the midpoint UP_v. (Note: this trade labeling method will be reconsidered and replaced later on.)

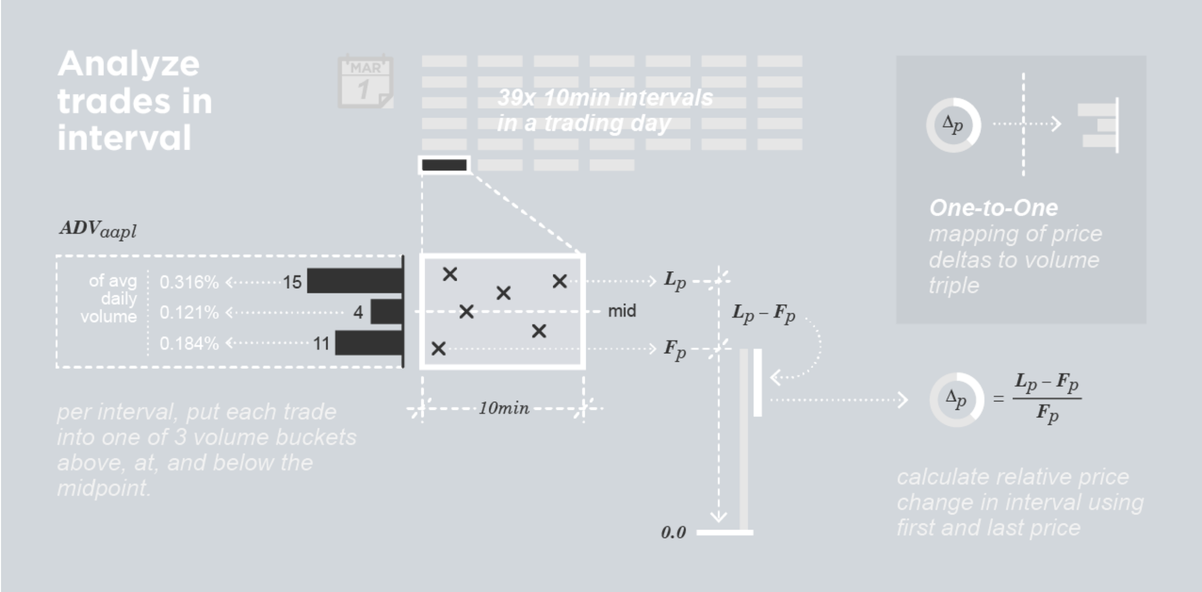
In these absolute terms, it is hard to meaningfully compare these volume numbers across symbols. What might be abnormally low amounts of trading in one symbol may be abnormally high in another. One way of dealing with this is to split all analyses by symbol, so that a time period of activity in one symbol is only ever compared to other time periods of activity in that same symbol. But this is a dramatic loss in sample size, as there are many thousands of symbols. So what we'll do instead is divide each volume by the average daily volume in that symbol to normalize it. [Note: we'll use a 20-day rolling average. More specifically, we look back 20 calendar days, and discard any days with no trading (e.g. weekends) before taking an average.]

After this normalization, our three numbers $UP_v, DOWN_v$, and $NEUTRAL_v$ for each time period are now percentages, and are typically less than 1 (though not always, as it is possible that a single 10 minute time period contains more volume than the average entire day). Since there are 39 10-minute time windows each trading day from 9:30 am to 4:00 pm, we would expect on average that our three numbers in each window will sum up to about $\frac{1}{39} \approx 0.0256$, though there will be lots of fluctuation.

In addition to these numbers intended to capture some basics of trading activity, we will also calculate the relative price change experienced over each time interval. More precisely, we will compute:

$$\Delta_p := \frac{L_p - F_p}{F_p},$$

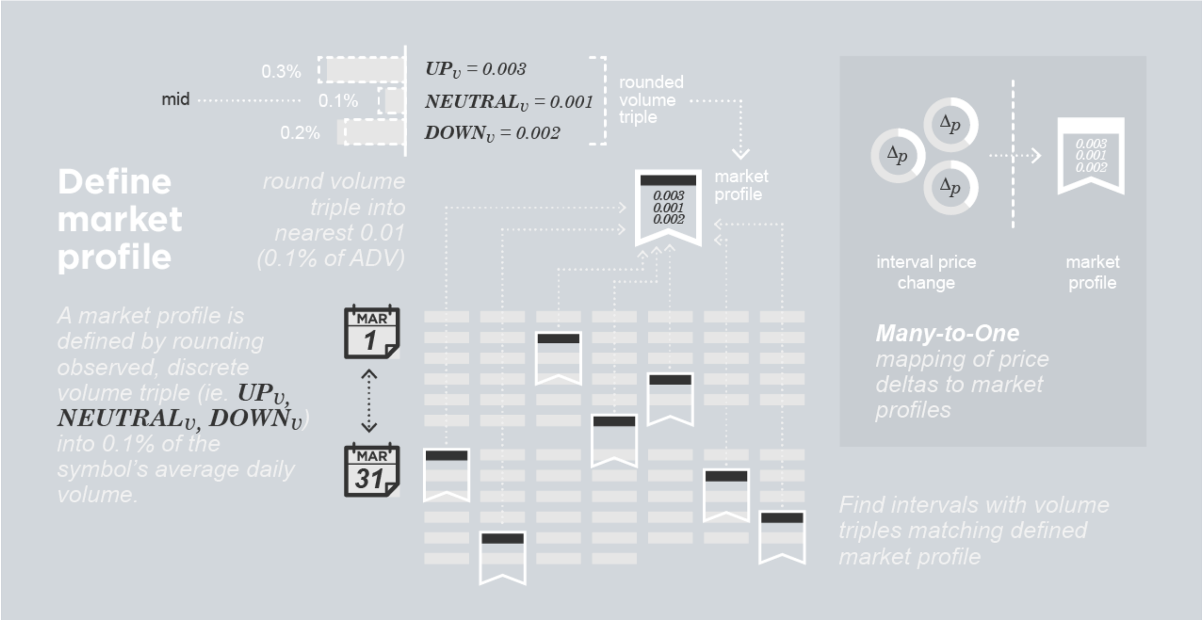
where L_p is the last trade price in the interval and F_p is the first trade price in the interval. Any intervals that lack trading activity entirely are discarded from the analysis.


Analyze trades in interval
MAR
1
39x 10min intervals in a trading day
ADV_{aapl}
of avg daily volume
0.316%
15
0.121%
4
0.184%
11
mid
L_p
F_p
10min
$L_p - F_p$
0.0
per interval, put each trade into one of 3 volume buckets above, at, and below the midpoint.
Δ_p
One-to-One mapping of price deltas to volume triple
$\Delta_p = \frac{L_p - F_p}{F_p}$
calculate relative price change in interval using first and last price

Naturally, we are interested in the relationship between our calculated triple (UP_v, $DOWN_v$, $NEUTRAL_v$) and the Δ_p values. This relationship is best modeled as probabilistic: each set of fixed values for $(UP_v, DOWN_v, NEUTRAL_v)$ induces a potentially different distribution on the random variable Δ_p. A priori, we don't know what these distributions are, but we have a collection of samples from them for various values of the market profiles, coming from historical data.

If we treat our three market profile components as having arbitrary precision and suppose that the distributions on Δ_p can change arbitrarily with even tiny changes to our market profile numbers, then we cannot hope to compute authoritative estimates of the Δ_p distributions from our data set. This is because we will have very few samples for each exact value of $(UP_v, DOWN_v, NEUTRAL_v)$ at high levels of precision. However, this is an overly pessimistic view. It is reasonable to assume that small changes to these numbers yield only small changes to the distribution of Δ_p in return.

There are two common approaches we can take to leverage this assumption. One would be to impose a limited form on the relationship between the market profile and Δ_p distribution. This is how techniques like regression work: we could for example, suppose that the expectation of the Δ_p distribution is a linear function of the market profile features, and find the best fitting linear function for a training set pulled from our historical data. Linearity here is a strong constraint though, so we'll start with the other approach: discretizing the market profiles. If we round each of the three numbers in our market profiles to a suitably coarse set of values that are qualitatively representative, we can group our observations by these rounded values and build up sample size in each group. For a suitably large group of observations that have the same (rounded) market profile, we don't have to impose any particular form on the relationship to a Δ_p distribution, we can just empirically measure it. We can estimate its expectation, for instance, by simply taking the average Δ_p in our group of samples, or we can estimate the probability it places on a given range of values by taking the proportion of our group of samples that land in that range.



Define market profile

A market profile is defined by rounding observed, discrete volume triple (ie. UP_v, $NEUTRAL_v$, $DOWN_v$) into 0.1% of the symbol's average daily volume.

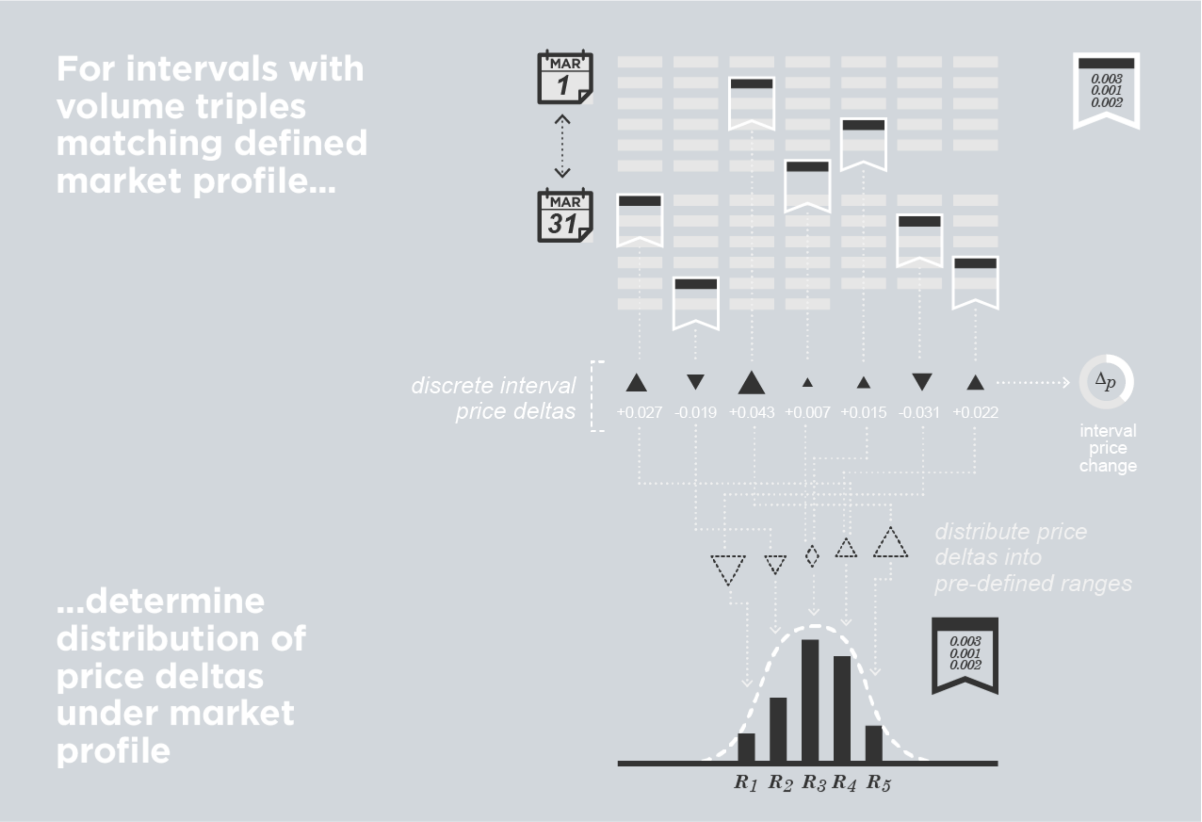
round volume triple into nearest 0.01 (0.1% of ADV)

Many-to-One mapping of price deltas to market profiles

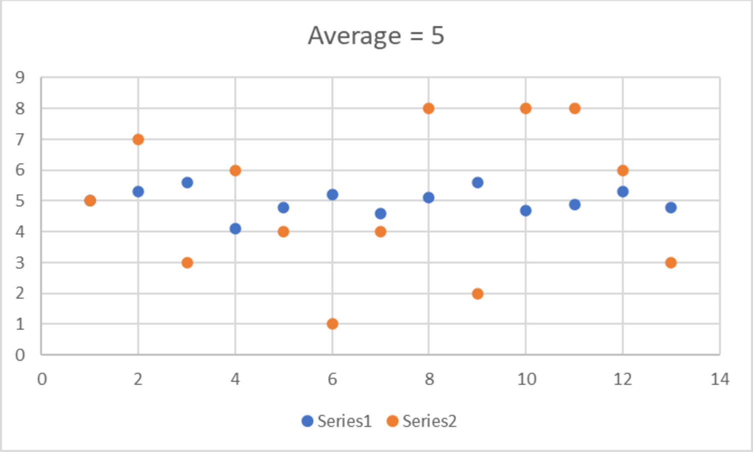
Find intervals with volume triples matching defined market profile

We will round our values for UP_v, $DOWN_v$, and $NEUTRAL_v$ to the nearest multiple of 0.01. Recall this unit represents 1 percent of the prior day's total volume. Since we expect the "average" sum of these three numbers to be around 0.0256, it may often be the case that all three numbers are rounded down to 0.0. This is fine, we just need to remind ourselves that numbers of 0 don't mean *no* volume, they just mean fairly low volume. In fact, about a third of our total observations fall into this category (for data from all symbols over the first quarter of 2019). This is probably much higher than we'd like and it would certainly be nice to increase our precision, say by rounding to the nearest multiple of 0.005 or even 0.0002. Doing this unilaterally would cause a proportionate reduction in sample size. We won't do this for now, as we'd prefer to start with a coarse model and make sure the sample size is sufficient to yield meaningful results, and then go back and try to refine it. Another option is to use something more like a logarithmic scale and increase the precision for very small numbers and decrease the precision as the numbers get larger. This would express the intuition that the difference between say 0.005 and 0.01 is more meaningful than the difference between 0.085 and 0.09. For now, we'll stick with the simplicity of 0.01 as a unit universally, but we'll keep it closely in mind that this is a highly problematic choice, and that what we're doing so far is basically a proof of concept only.

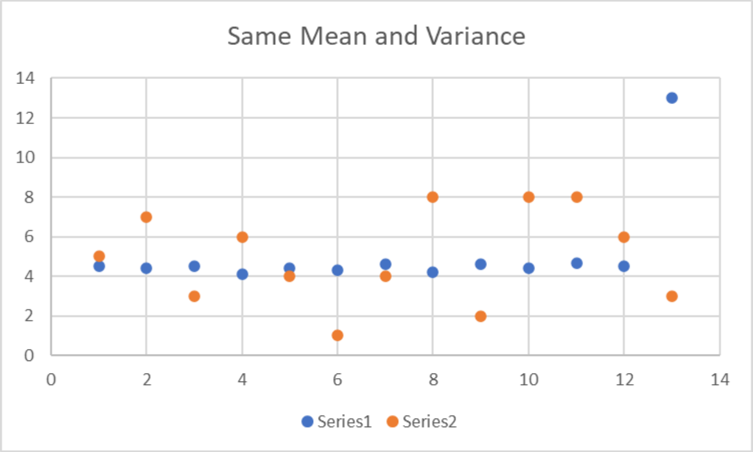
Next we have to decide what we want to measure about the (hopefully many) observations of Δ_p that we have for each rounded market profile. Initially, what we have is a collection of hundreds or thousands of individual values for Δ_p that come from hundreds or thousands of times we have observed a particular market profile in our historical data set.


For intervals with volume triples matching defined market profile...
MAR 1
MAR 31
0.003
0.001
0.002
discrete interval price deltas
+0.027
-0.019
+0.043
+0.007
+0.015
-0.031
+0.022
Δp
interval price change
distribute price deltas into pre-defined ranges
...determine distribution of price deltas under market profile
0.003
0.001
0.002
R_1 R_2 R_3 R_4 R_5

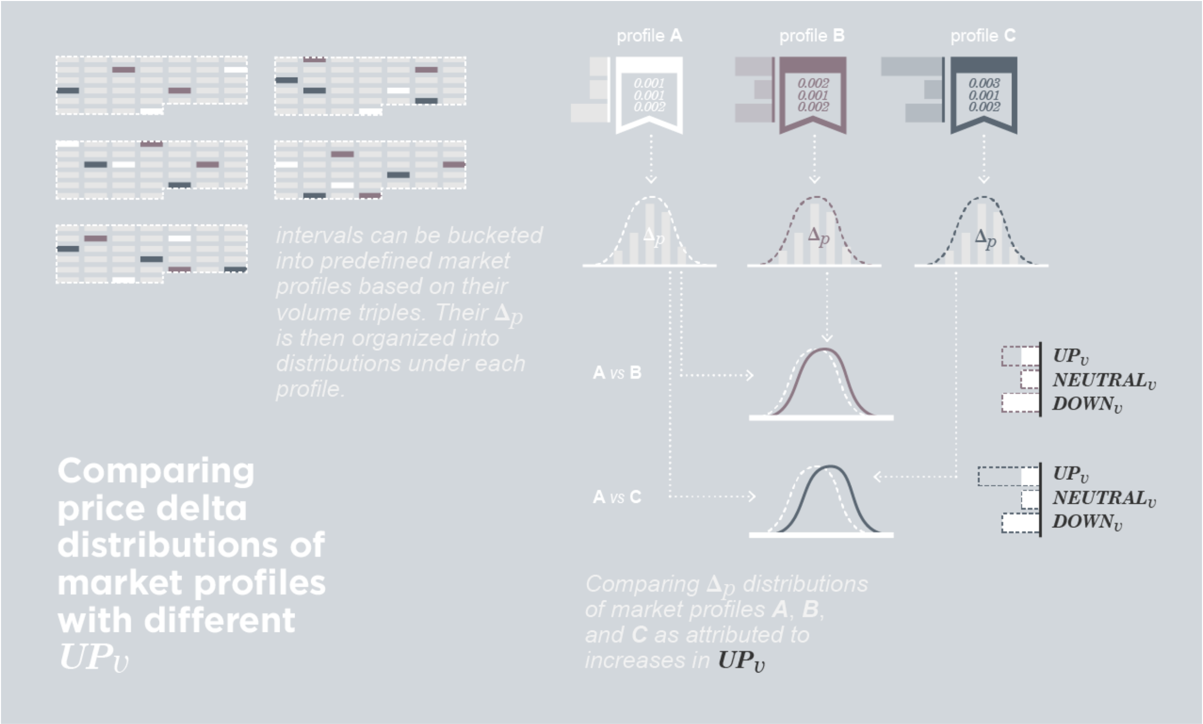
We could collapse this collection of data to a single number by taking its raw average, which would be our empirical estimate of the expected value for Δ_p associated with this particular market profile. But we may be ignoring a lot of useful information when we do this. For example, here are two data series that have the same average value, but the blue series is far more concentrated around its average than the orange series:



There are many interesting ways to summarize a distribution that fall in between the extremes of carrying around the full set of observations and collapsing it to a single average. One common approach is to compute the *variance* or *standard deviation* of the distribution, and add this to the average to paint a somewhat fuller picture. This approach conveys most of the relevant information in cases where the distribution is particularly well-behaved, meaning that it is fairly bell-shaped (Gaussian). However, it is less informative in cases where the distribution has considerable outliers that exert a heavy pull on the average as well as variance. Note the variance is itself an average (of the square of the deviation from the mean), and hence it may also be heavily influenced by outliers. Here are two series that have the same mean and variance, even though the blue series variance is mostly coming from a single outlier:



If we want to concisely capture what's important about our Δ_p samples and the underlying distributions they represent, we have to ask: what do we most want to know about these distributions, and what sorts of misbehavior might overly skew our answers? In terms of what we want to know, one important thing is: how *different* are the Δ_p distributions? In other words, how much influence do market profiles have on the distributions of relative price fluctuations? Fundamentally, this requires us to compare distributions. Given samples from two distributions, we want to measure how similar the distributions might be.



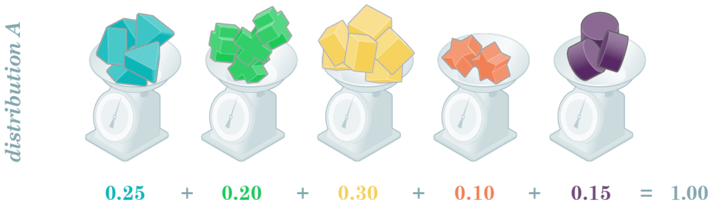
intervals can be bucketed into predefined market profiles based on their volume triples. Their Δ_p is then organized into distributions under each profile.

Comparing price delta distributions of market profiles with different UP_v

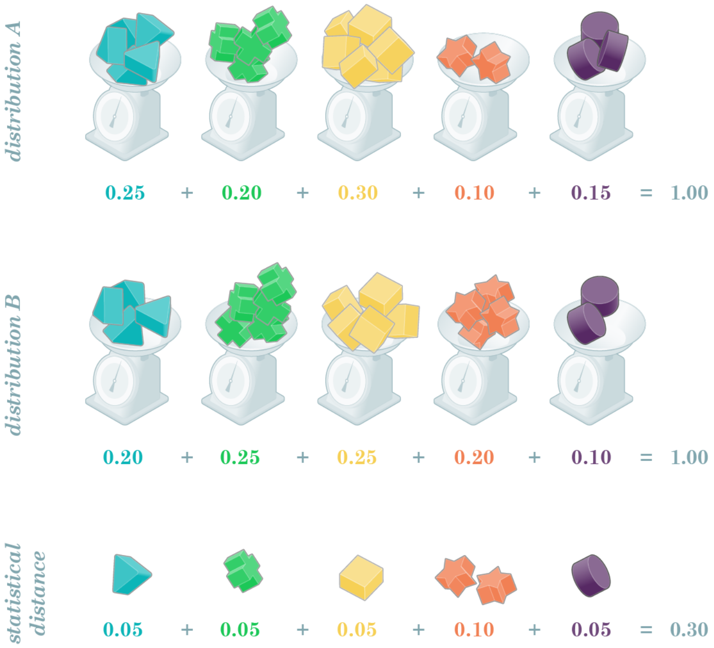
Comparing Δ_p distributions of market profiles ***A***, ***B***, *and* ***C*** *as attributed to increases in* $\boldsymbol{UP_v}$

Comparing means and variances can get us started, but we should be wary that financial data often has heavy tails, and outliers may therefore cause two sets of samples to look artificially different when they are (mostly) similar. We could screen for this kind of effect by testing how similar our distributions are to well-behaved Gaussian distributions, but then designing the screening procedure presents the same problem! We've just moved the task of measuring similarity of two arbitrary distributions to the task of measuring the similarity of one arbitrary distribution and one Gaussian distribution.

Thus it seems helpful to use a more general way of measuring similarity of distributions, and for this we will use a notion of *statistical distance*. There are many reasonable notions of statistical distance, but we'll stick with a pretty basic and intuitive one. To define our notion of statistical distance, we'll start with the toy case of distributions on just 5 possible values. We can think of each value as a scale, and the probability of that value is represented by the weight of objects we place on top of that scale.



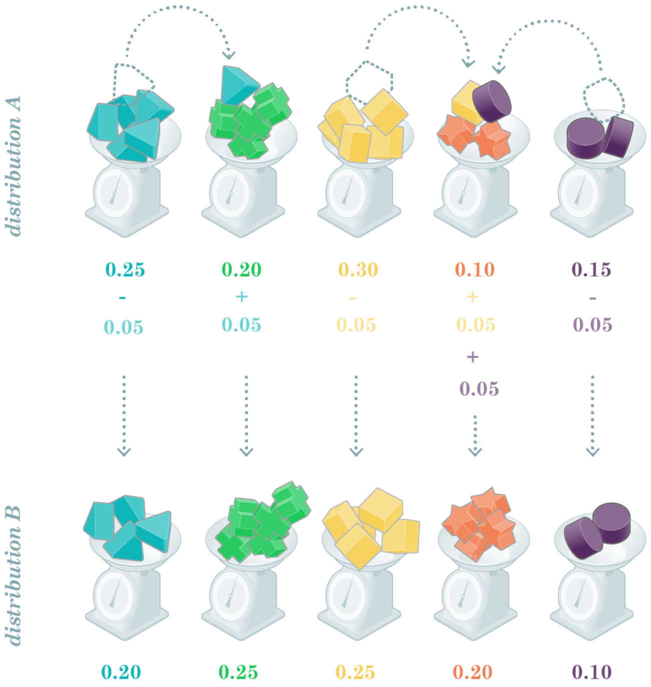
The weight is measured in units that are normalized so that the total weight summed across all 5 scales equals 1. Now, if we have two such distributions, we can define the statistical distance between them as the sum of the absolute values of the differences in the weights on each scale:



This variant of statistical distance is known as *total variation distance*. We'll let $SD(D_A, D_B)$ denote this distance between distributions D_A and D_B. We can generalize this notion easily to distributions over any finite set of discrete values. To write this formally, we'll let U denote our finite universe of possible values, and for each value $u \in U$, we'll let $w_A(u), w_B(u)$ denote the weights that distributions D_A and D_B place on value u respectively. Then:

$$SD(D_A, D_B) := \sum_{u \in U} |w_A(u) - w_B(u)|.$$

This represents twice the amount of weight we need to move in order to turn one distribution into the other. This factor of 2 comes from the fact that moving weight from one scale onto another affects the absolute value differences for *both* scales:



This intuitive physical meaning is one nice feature of using this statistical distance to compare distributions. Another nice feature is relative robustness to outliers: small weights have only a small effect on total variation distance, even if they are placed on extreme values. Another nice feature is symmetry: $SD(D_A, D_B) = SD(D_B, D_A)$, so it doesn't matter which distribution we call "D_A" and which distribution we call "D_B." This is a typical property we require of notions of "distance": the distance between two objects should not depend on which one is considered the starting point. Similarly, we have $SD(D_A, D_A) = 0$, so the distance of a distribution from itself is 0. Also we have the triangle inequality:

$$SD(D_A, D_C) \leq SD(D_A, D_B) + SD(D_B, D_C).$$

This states that the distance between D_A and D_C is no greater than the sum of the distance between D_A and D_B and the distance between D_B and D_C.

Of course, we do not know the "true" Δ_p distributions for each market profile, we only have our observed data. So to estimate things like total variation distance, we will need to estimate the weights that our distributions place on various values. If we define our universe of values to be our calculated values of $\frac{L_p - F_p}{F_p}$ at a high level of precision, we'll once again face the problem of having insufficient sample size per value to get a reliable estimate of its weight. So we'll need to group together ranges of values of $\frac{L_p - F_p}{F_p}$ in order to get sufficient sample size to estimate the probability mass that falls within each range.

But how should we choose our ranges? We don't want to have too many, as this will again lead to small sample sizes in each range. We probably also want to avoid choosing ranges so that the weight tends to concentrate unevenly in them, as heavier ranges might indicate we are grouping together values that we could afford to distinguish, and lighter ranges might suffer inaccurate estimates. With this in mind, what we'll do first is pick a number of ranges we'd like to have, let's say 21. Next we'll look across a sizeable sample of our data set of Δ_p calculations and order it from most negative to most positive. We'll find break points that divide it into 21 equal pieces (e.g. this would be dividing it into quartiles if we were doing 4 ranges instead of 21). These breakpoints will become the delineators of our ranges, with one small adjustment. Since Δ_p distributions are likely to exhibit a lot of symmetry around 0, we'll make sure our middlemost range is narrowly centered on 0, and we'll have 10 ranges on either side of it. (This is why we choose an odd number of ranges to begin with.) It's helpful for us to avoid choices of ranges like $[-0.001, 0)$ and $[0, 0.001)$, since putting all of the calculations that are rounded to 0 on the "positive" side will destroy expected symmetries between positive and negative price movements.

Once we have defined these ranges, we will apply them to summarize Δ_p distributions for *all* market profiles, so that we can use these apples-to-apples range definitions to meaningfully compare different Δ_p distributions corresponding to different market profiles. So for each market profile, we'll take all our observed Δ_p values, and we'll count what proportion of them fall in each one of our defined ranges. This gives us 21 numbers (adding up to 1) which summarize our distribution, and enable us to compute the statistical distance between two such distributions with respect to these fixed ranges. We'll separately compute the average value of the observations for each Δ_p distribution as well, since we may independently want this, and estimating it from the data after we've rounded to ranges would introduce an unnecessary source of error.

Putting this all together, we're ready to grab a bunch of historical data and compute summaries of the Δ_p distributions corresponding to rounded market profile values. What we'll get is a big table of numbers like these:

UP_v	$DOWN_v$	$NEUTRAL_v$	$\mathbb{E}(\Delta_p)$	$\mathbb{P}(\Delta_p \in R_1)$	$\mathbb{P}(\Delta_p \in R_2)$	$\ldots$	$\mathbb{P}(\Delta_p \in R_{21})$
0	0	0	-0.00001	0.03331	0.03722	$\ldots$	0.03390
0	0	0.01	$+0.00002$	0.02046	0.03495	$\ldots$	0.01964
0	0.01	0	-0.00085	0.02423	0.02617	$\ldots$	0.05631
0.01	0	0	$+0.00081$	0.05257	0.06926	$\ldots$	0.02216
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ldots$	$\vdots$

If you're not familiar with the notation here, don't worry - we'll break it down for you. $\mathbb{E}$ is math speak for "expectation," so $\mathbb{E}(\Delta_p)$ denotes the average value of the random variable Δ_p. We're using $R_1, \ldots, R_{21}$ to denote our 21 ranges of values, and the notation $\mathbb{P}(\Delta_p) \in R_i)$ denotes the probability that Δ_p takes on a value inside the range R_i. (Of course, it's important to remember that our table contains empirically computed estimates, not ground truth probabilities.) R_1 contains the highest values of Δ_p, and R_{21} contains the lowest values of Δ_p.

We are able to do this sufficiently quickly - we can assemble data from all symbols over the span of 3 months at a time in about one hour. Hooray! But two major questions remain: 1) how do we know these numbers are meaningful? and 2) what do we do with them? We'll be exploring these questions in the following sections.

4 Sanity and robustness checks

Let's start with the question of how we might assess the quality of the numbers we've computed. There are multiple ways things could have gone wrong. Here's just a small sample of them: 1. perhaps our estimates of the averages and probability masses in each range for many Δ_p distributions are wildly different from the true values. 2. perhaps the Δ_p distributions don't vary much by the market profiles, and hence this data is not very interesting. 3. perhaps the true Δ_p distributions change on a timescale that is shorter than our 3 month training data set, and hence this information will not replicate going forward.

There are some sanity checks we can perform to gain confidence that these kind of failures are not destroying our results. The first check we will perform will test how stable these estimated distributions are over time. For this, we'll compare the results of estimating our Δ_p distributions from data for January through March of 2019 to the results of doing the same thing from data for April through June of 2019. For each market profile that appears in both data sets, we can compute the statistical distance between our two estimated distributions. We'll average these statistical distances over the market profiles, weighting each by its prevalence in our Jan-March 2019 data set. [Note that some market profiles may occur in one data set and not the other. We have checked that these represent only a small amount of the total probability mass, and thus we will ignore this minor issue.] We get an "average" statistical distance of approximately 0.06. This means, on average, about 3% of the probability mass moves around relative to our fixed Δ_p ranges when we use one data set versus the other. (Of course, for some market profiles the difference will be less, and for some it will be greater.)

We can also ask: how does the relative prevalence of each market profile change from one data set to the other? For this, we can compute the absolute value of the weight difference for each market profile and sum up over all the market profiles. We get a number that is approximately 0.04. Combined, these checks tell us something about the net effect of noise and time-drift effects on our estimates. Overall, it's an encouraging picture so far, as errors on these orders are likely to be survivable and not destroy overall meaning if we are careful.

Next, we will check if our distributions (for Jan-March 2019) are roughly similar to each other, or if they vary considerably as the market profiles vary. To do this, we'll compute an "average" distribution and then compute the statistical distance from each distribution to the average. To average the distributions, we'll individually average the probability mass placed in each of our ranges, and we'll weight according to the prevalence of each market profile. Once we have the average distribution and the statistical distance from it for each distribution, we'll again average these statistical distances, weighted by profile prevalence, to get a single number that represents how far our distributions typically are from the average. If all of our distributions were the same, we would expect this value to be close to the 0.06 that may represent the level of noise in our estimates. But instead, we get a much larger value: approximately 0.24. This is all good evidence that our Δ_p distributions do vary considerably by market profile, and they do so in ways that are stable enough over months-long time periods to get reasonable estimates.

Another interesting check we can perform is a symmetry check: does the Δ_p distribution for a market profile like $UP_v = 0.06, DOWN_v = 0.01, NEUTRAL_v = 0.02$ behave similarly to the Δ_p distribution for the market profile $UP_v = 0.01, DOWN_v = 0.06, NEUTRAL_v = 0.02$, but with the signs reversed? In other words, if we swap the UP_v and $DOWN_v$ values in a market profile, leaving $NEUTRAL_v$ fixed, can we just swap the signs of Δ_p distribution? This is something we would expect to work, as it is intuitive to guess that downward pressure on prices behaves the same as upward pressure, just with the sign of the price change reversed. However, general market trends in our data set may perturb this, as there may be a default tendency for prices to go up (or go down) over the time period of our data collection.

To perform the symmetry check, we'll match up all of the pairs of distributions with $UP_v, DOWN_v$ swapped, flip the sign of one of the Δ_p distributions, and compute the statistical distance. Note that we choose our ranges for the Δ_p values to be roughly symmetric around 0, so this is easily approximated by swapping the weights of ranges 1 and 21, the weights of ranges 2 and 20, etc. Here is an example where the first row is for $UP_v = 0$, $DOWN_v = 0.02$, $NEUTRAL_v = 0.01$, and the second row is for $UP_v = 0.02$, $DOWN_v = 0$, $NEUTRAL_v = 0.01$:

$\mathbb{E}(\Delta_p)$	R_1	R_2	R_3	R_4	...	R_{11}	...	R_{18}	R_{19}	R_{20}	R_{21}
-0.0016	0.017	0.022	0.022	0.026	...	0.101	...	0.065	0.079	0.094	0.097
$+0.0015$	0.094	0.088	0.070	0.063	...	0.107	...	0.023	0.022	0.020	0.017

We've condensed our notation $\mathbb{P}(\Delta_p \in R_i)$ to simply R_i here to save horizontal space in our display.

Let's see those R_i weights again with the bottom row flipped around the R_{11} range centered on 0:

0.017	0.022	0.022	0.026	...	0.101	...	0.065	0.079	0.094	0.097
0.017	0.020	0.022	0.023	...	0.107	...	0.063	0.070	0.088	0.094

It's not perfect of course, but we can certainly see the resemblance.

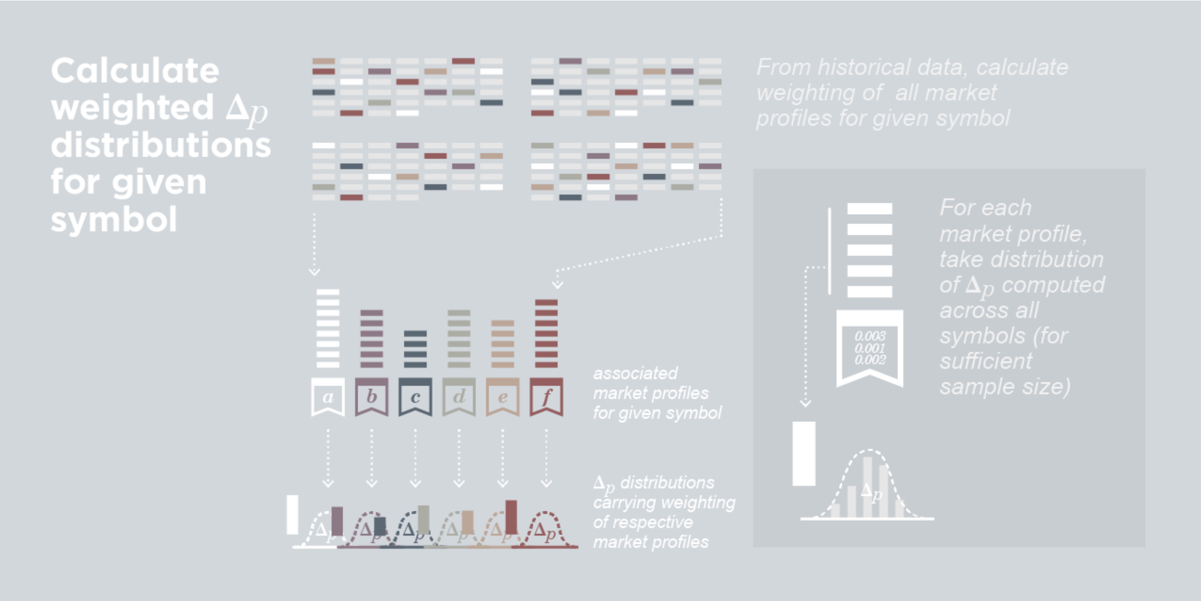
When we do this for all of our data and average the resulting statistical distances according to market profile prevalence, we obtain a value that is approximately 0.05.

Putting all this together, we have good evidence that our estimated Δ_p distributions are roughly stable over months, roughly symmetric as we would expect, and display a meaningful dependence on market profiles.

5 Prediction with a single time unit

Now we have gotten to a point where we have a reasonably effective grasp on how coarsely rounded market profiles correspond to distributions of Δ_p over 10 minute intervals. How might we leverage this into a predictive model of what will happen if we try to trade a certain amount of volume over a 10 minute interval? To make things concrete, let's suppose that we intend to buy an amount of MSFT stock over 10 minutes that corresponds to 2% of the average daily volume. Let's suppose for now that we will cross the spread to do so, which means we'll be adding $+0.02$ to the variable UP_v.

Before we decide to take action, the market profile for any 10 minute period in MSFT can be treated as a random variable, and we can get a sense of its distribution by looking at historical data in MSFT and measuring the relative prevalence of various market profiles. This gives us a weighting over market profiles that we associate with the symbol MSFT.


Calculate weighted Δ_p distributions for given symbol
From historical data, calculate weighting of all market profiles for given symbol
For each market profile, take distribution of Δ_p computed across all symbols (for sufficient sample size)
0.003
0.001
0.002
Δ_p
a
b
c
d
e
f
associated market profiles for given symbol
Δ_p distributions carrying weighting of respective market profiles
Δ_p

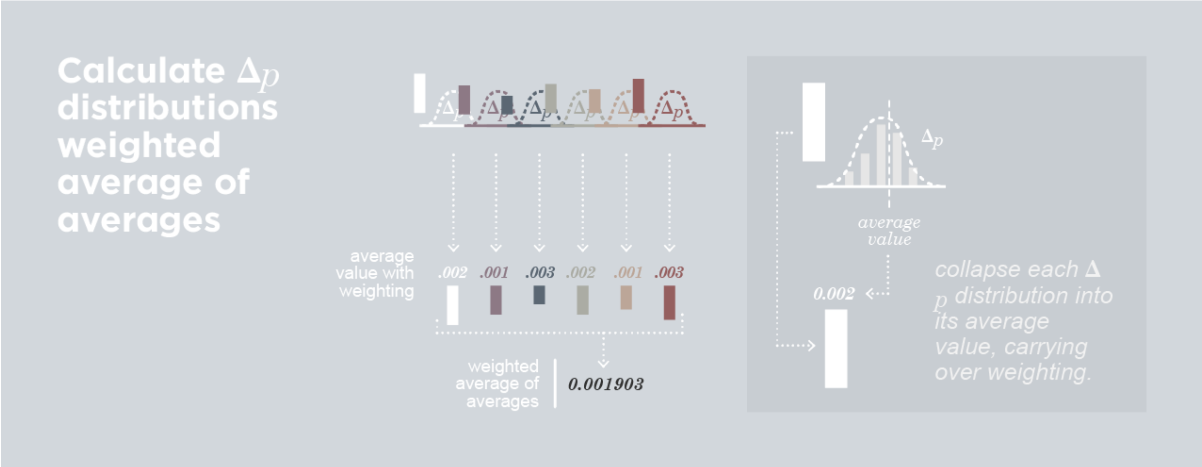
For each market profile, we already have an estimated distribution of Δ_p (computed across all symbols to achieve sufficient sample size). Putting these things together, we have a weighting of Δ_p distributions that represents our default expectation for what should happen to Δ_p for MSFT over a 10 minute time interval.

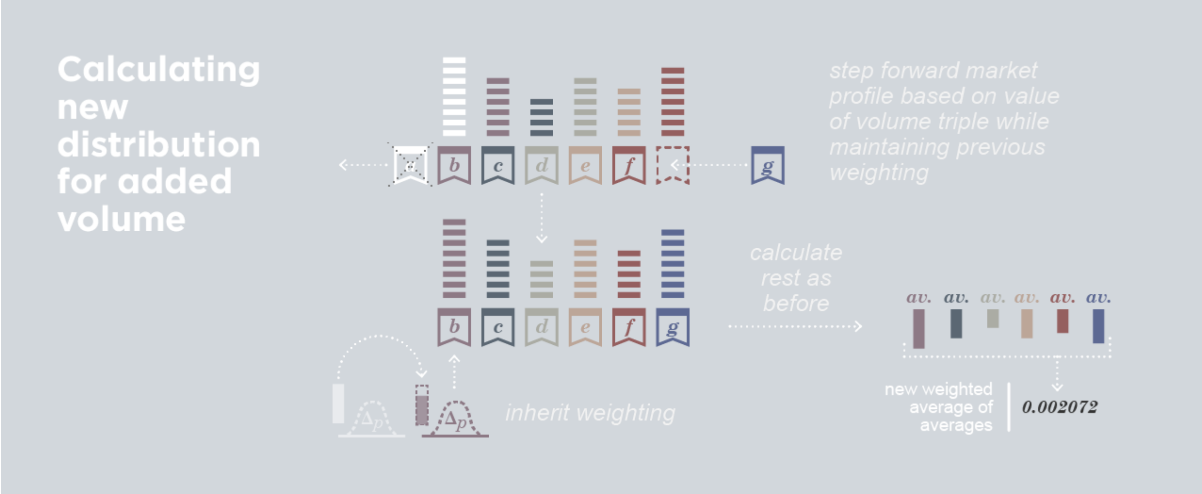
We note that it could be valuable to do this computation separately for different times of day: perhaps the typical market profile distribution for MSFT from 10:00 am to 10:10 am looks different than the typical market profile distribution for MSFT from 3:00 pm to 3:10 pm. Trying to account for this will naturally shard our data set into finer pieces, leaving less sample size in each piece. One way to mitigate this is to try to identify times that are roughly similar to each other, so that we can find a happy medium between treating all 10 minute intervals as the same and treating all of them as completely different. The same approach could be applied to sharding by symbol. This is a delicate issue we will come back to, but for now we'll build a baseline by throwing the 10-minute intervals of one symbol all together and computing one single distribution of market profiles for each symbol, averaging evenly over time of day.

Our action of adding +0.02 to UP_v an be viewed as a 1-1 mapping between market profiles. A market profile of $(UP_v = 0.03, DOWN_v = 0.02, NEUTRAL_v = 0.02)$ without our added volume becomes a market profile of $(UP_v = 0.05, DOWN_v = 0.02, NEUTRAL = 0.02)$ with our added volume. We can take these new market profiles (with +0.02 added to the UP_v) and treat them as inheriting the weight of their pre-images under this mapping. In other words, we assume that with our activity accounted for, the market profile of $(UP_v = 0.05, DOWN_v = 0.02, NEUTRAL = 0.02)$ now occurs with the probability that was originally associated with the market profile of $(UP_v = 0.03, DOWN_v = 0.02, NEUTRAL_v = 0.02)$. This gives us a new weighting on market profiles that shifts weight towards higher values of UP_v. Again invoking our pre-computed estimates of Δ_p distributions per market profile, this gives us an associated weighting on Δ_p distributions that represents our guess for what behavior we expect in price movements, given our intended actions.

Of course, this does not fully capture reality. In live trading, any insertion of activity can cause a myriad of reactions to that activity, and it's possible that the end result will be quite different from a simple addition of this type. This is something we could test if we were already live trading: does the distribution of the resulting market profile after our 10 minutes of activity look like the expected additive shift in UP_v? This is also something anyone can test in TCA analysis of historical training data. (All we really need is for the trading activity of one party to be labeled so that we can separate it from the rest of the market.) What we learn from real trading data may inspire us to refine this aspect of the model over time, and to customize it to account for different tactics for trading within a time window. For now, since we are using public historical data that does not have trades attributed to parties, we will stick with this base assumption that the effect of our activity on the market profile is a simple addition. [Spoiler alert: later in this paper, we will settle on a variant that is still a simple addition, but added to a variable that is defined differently from UP_v.]

Next, we want to collapse these weightings on estimates of Δ_p distributions into concise, digestible numbers we can more easily look at. For this, we'll first collapse each Δ_p distribution to its average value, and then compute the weighted average of these averages. What we get is a single number that summarizes our expected price impact.


Calculate Δ_p distributions weighted average of averages
Δ_p
average value with weighting
.002
.001
.003
.002
.001
.003
weighted average of averages
0.001903
Δ_p
average value
0.002
collapse each Δ_p distribution into its average value, carrying over weighting.


Calculating new distribution for added volume
b
c
d
e
f
g
step forward market profile based on value of volume triple while maintaining previous weighting
calculate rest as before
av.
av.
av.
av.
av.
av.
new weighted average of averages
0.002072
Δ_p
inherit weighting

It is important to remember that we are losing a lot of information here by looking only at this average, but it's a reasonable place to start. It's also important to keep track of the sample sizes that our numbers are based on, so we know to treat numbers based on smaller samples with increased skepticism. We'll use our same weightings to compute an "average" sample size for each average Δ_p value we compute. Keep in mind that we are taking a weighted average of the sample sizes across all symbols used to estimate our Δ_p distributions, not the sample sizes for our symbol-specific computation of the weights themselves.

We can compute these numbers for different symbols and for different amounts of volume that we may plan to buy. Here's what we get, for instance, for MSFT:

Added Volume	Estimated Impact	Avg Sample Size
0	0.0000098	1503370
0.01	0.0009910	420047
0.02	0.0014564	75737
0.03	0.0016782	31927
0.04	0.0018048	17581
0.05	0.0017999	11047
0.06	0.0019435	7491
0.07	0.0020908	5406
0.08	0.0020881	4045
0.09	0.0019785	3171
0.10	0.0020859	2509

As the amounts of volume we intend to trade increase, our numbers become more precariously based on smaller sample sizes that may become more and more correlated with unusual market conditions, since higher amounts of volume trading in 10 minute intervals are not too common. We can clearly see that phenomenon here, and we should expect that our estimated impact becomes less and less meaningful as a result.

There are a few things to note about these numbers. First, the fact that the estimates of impact are not perfectly monotonically increasing is likely an artifact of noise and declining sample size. But clearly, there is a trend of fast growth, and then it slows and/or devolves into noise and correlated effects.

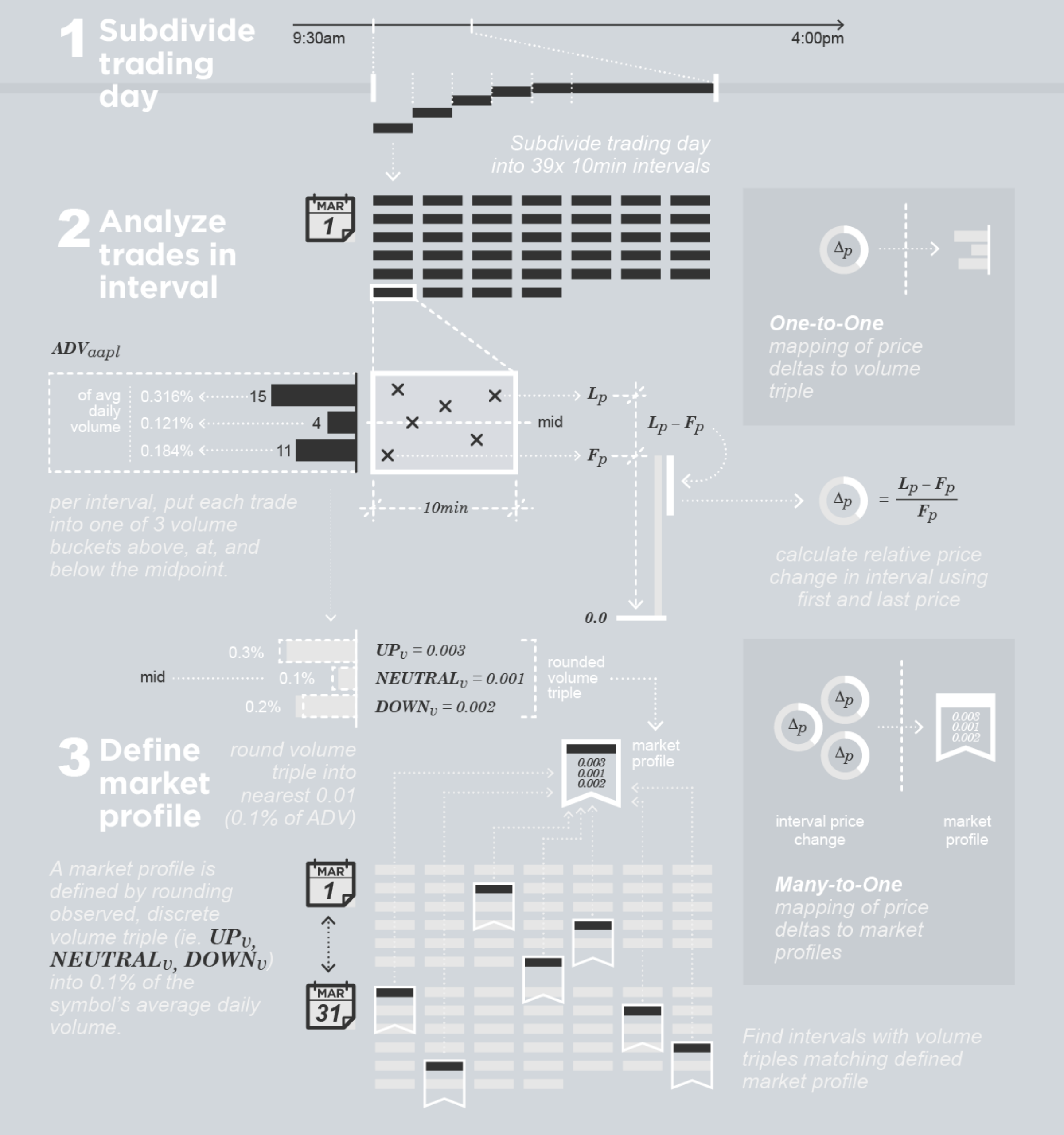
Similarly, here's what we get for BAC:

Added Volume	Estimated Impact	Avg Sample Size
0	−0.0001170	1311001
0.01	0.0008864	434554
0.02	0.0014896	73719
0.03	0.0017716	29241
0.04	0.0019473	15661
0.05	0.0019847	9673
0.06	0.0020484	6486
0.07	0.0022630	4631
0.08	0.0023015	3435
0.09	0.0022287	2668
0.10	0.0023093	2099

There doesn't seem to be much deviation per symbol here. This suggests that we may want to group symbols by similar behavior rather than computing the relative prevalence of market profiles for each individually. Additionally, it suggests that to the extent that meaningful deviation between symbols exists, it might reside in the link between market profiles and Δ_p distributions, rather than the relative prevalence of market profiles.

But that's a problem for another day. Let's look back on what we've done so far, and draw a full picture of the computational process we've developed:

DEFINING UNITS & MARKET CONDITIONS



1 Subdivide trading day

Subdivide trading day into 39x 10min intervals

2 Analyze trades in interval

per interval, put each trade into one of 3 volume buckets above, at, and below the midpoint.

$$\Delta p = \frac{L_p - F_p}{F_p}$$

calculate relative price change in interval using first and last price

One-to-One mapping of price deltas to volume triple

3 Define market profile

round volume triple into nearest 0.01 (0.1% of ADV)

$UP_v = 0.003$

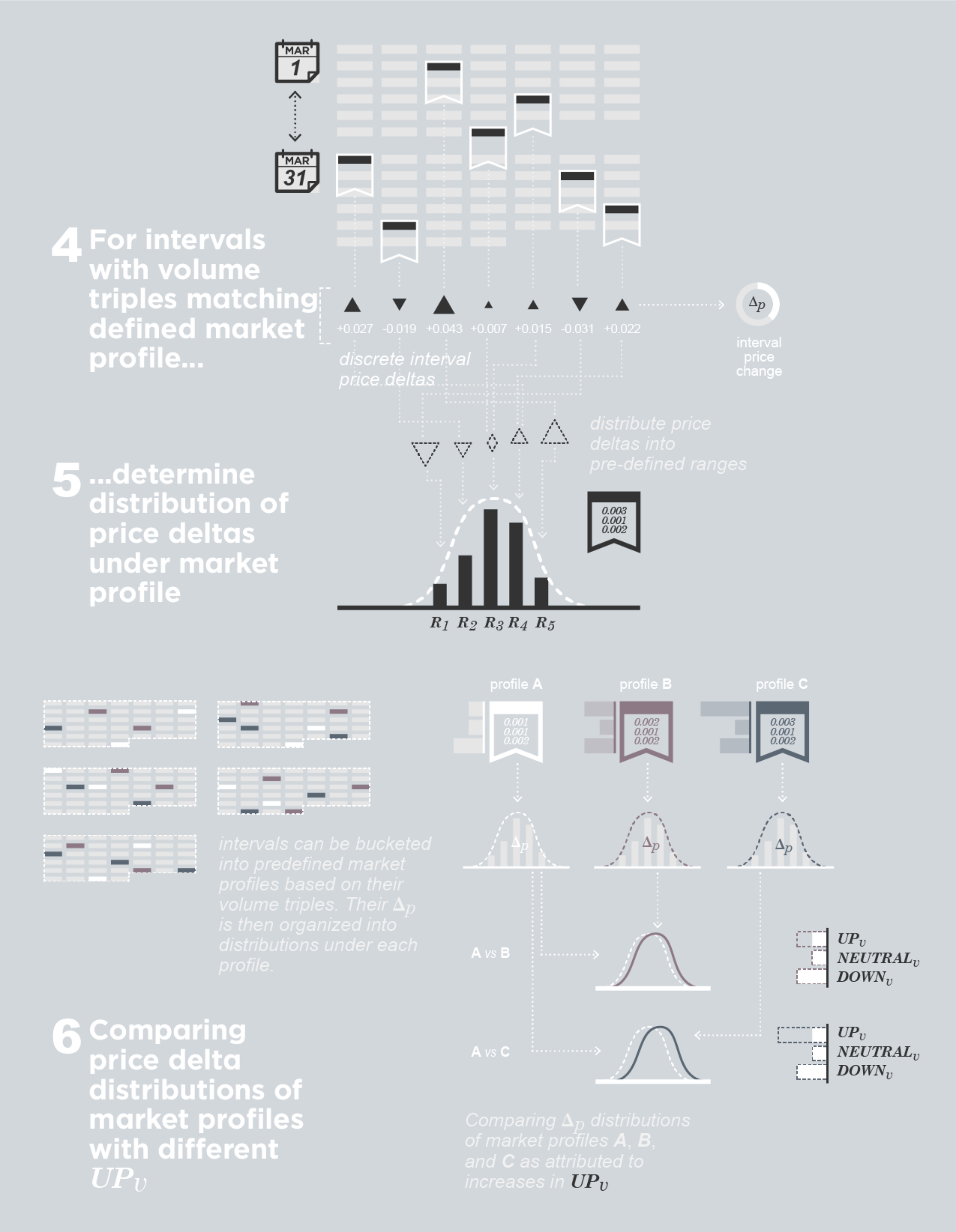
$NEUTRAL_v = 0.001$

$DOWN_v = 0.002$

A market profile is defined by rounding observed, discrete volume triple (ie. UP_v, $NEUTRAL_v$, $DOWN_v$) into 0.1% of the symbol's average daily volume.

Many-to-One mapping of price deltas to market profiles

Find intervals with volume triples matching defined market profile



4 For intervals with volume triples matching defined market profile...

discrete interval price deltas

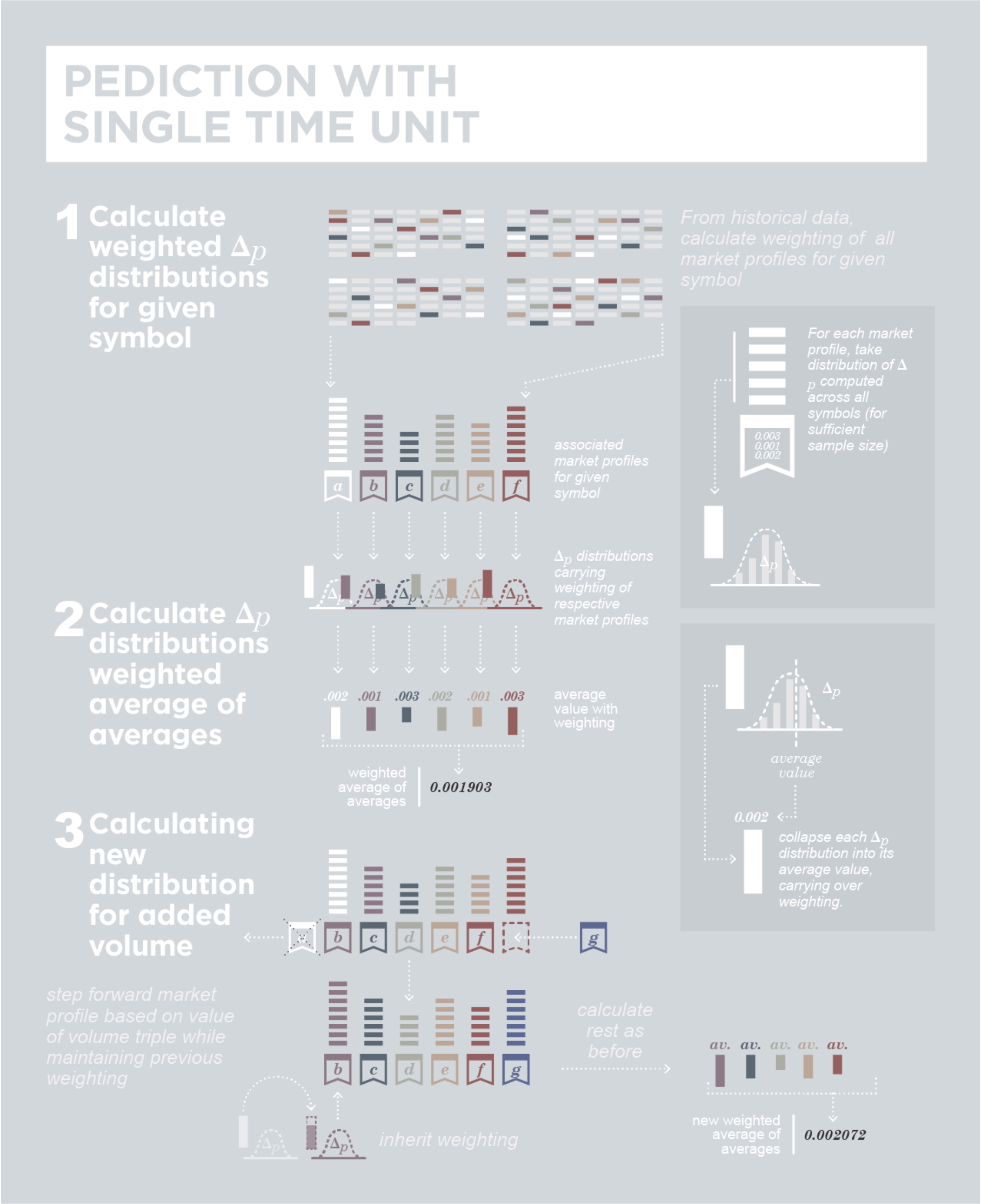
5 ...determine distribution of price deltas under market profile

distribute price deltas into pre-defined ranges

intervals can be bucketed into predefined market profiles based on their volume triples. Their Δ_p is then organized into distributions under each profile.

6 Comparing price delta distributions of market profiles with different UP_v

*Comparing Δ_p distributions of market profiles **A**, **B**, and **C** as attributed to increases in **UP_v***



PEDICTION WITH SINGLE TIME UNIT

1 Calculate weighted Δ_p distributions for given symbol

From historical data, calculate weighting of all market profiles for given symbol

For each market profile, take distribution of Δ_p computed across all symbols (for sufficient sample size)

associated market profiles for given symbol

Δ_p distributions carrying weighting of respective market profiles

2 Calculate Δ_p distributions weighted average of averages

average value with weighting

collapse each Δ_p distribution into its average value, carrying over weighting.

weighted average of averages 0.001903

3 Calculating new distribution for added volume

step forward market profile based on value of volume triple while maintaining previous weighting

inherit weighting

calculate rest as before

new weighted average of averages 0.002072

And now we can make some predictions! As long as we stick to trading that we expect to wholly accomplish within 10 minutes, and we stay inside these volume thresholds that have healthy sample sizes, and we caution ourselves to remember all of the assumptions and caveats we've made above and ... ugh. This is starting to sound like the side effect list at the end of a pharmaceutical ad. This is a bit deflating, but fear not! There are things we can do to start lifting some of these restrictions.

The most basic one is the restriction to a 10 minute time interval. We can repeat all of our steps above for any length of time interval, say 1 hr, or 1 day. There is a catch though:

longer time intervals mean fewer observations. For every trading day, there are 6.5 trading hours (excluding pre and post market sessions for now). And for every hour, there are 6 ten minute intervals. This means that as we jump from 10 minute intervals to hours and then again from hours to days as our fundamental unit of time, we are loosing sample size by a factor of at least 6 at each jump. If we want to combat this loss of data by saying, using a training data set from a time period that is 6 times as long, we may run into problems. To get comparable sample size, we'd need to hope that trends that were roughly stable on three month timescales for 10 minute periods extend to trends that are roughly stable on 18 month timescales for 1 hour periods, and this is already straining our total data availability to test, as we have slightly less than 3 years worth of total historical data on hand.

There are at least two possible ways out of this dilemma. One is to try to train with less than 6 times the amount of data for 1 hour increments and see if we can still get estimated Δ_p distributions that pass our robustness checks explained above. We can roughly binary search: if 3 months of data proves insufficient for studying 1 hour intervals, perhaps 6 months of data will be enough. If not, perhaps 12 months, etc.

A second possibility will be discussed in the next section: viewing these larger time intervals as sequences of smaller time intervals. This requires us to imagine how our individual choices of how much to trade in each smaller interval will combine to effect our overall impact.

6 Prediction with simulated choices across time units

Let's imagine now that we want to buy an amount of MSFT stock that represents 6% of the total volume from the previous day, and we want to accomplish this over the next twenty minutes. There are many ways we could split this into amounts to buy in each of the 10 minute intervals. We could buy it all in the first interval, or all in the second. We could split it evenly, adding +0.03 in each interval. We could split it unevenly, adding say +0.02 in the first interval and +0.04 in the second.

If we intend to trade wholly inside the first interval or wholly inside the second, we already have our estimate of the price impact. So let's consider how we might estimate the price impact of a proposed split, say +0.02 and +0.04. We have an estimate for the effect of our trading in the first interval - our expectation is that the price will be 0.0015 percent higher by the end of the first interval than it was at the beginning. (Note here that we use "expectation" in the formal mathematical sense of average, and the true value will have substantial variation around this mean.) We might also apply our 10-minute model above to estimate that the additional price movement in the second interval will be +0.0018 in expectation, now *starting from the higher baseline expectation for the end of the first interval.* Putting this together, we might suppose the final price at the end of the twenty minutes, relative to the starting point, will be in expectation: $1.0015 * 1.0018 \approx 1.0033$. This is a total expected Δ_p of +0.0033 for this 20-minute interval.

Since our estimate here depends upon our choice of how to split our volume over the two pieces, it would make sense to choose the split that minimizes the total expected impact. Doing this would allow us to assign a single expected impact value to a twenty minute interval. We could then derive estimates for 30 minute intervals by taking the minimal impact estimate over all ways of splitting the volume across a 10-minute interval and a 20-minute interval. Once we have estimates for 30 minute intervals, we could continue with 40-minute intervals and so on. This technique is called *dynamic programming.* It consists of re-formulating larger problems (in our case, estimates for longer intervals) into sets of smaller problems (estimates for shorter intervals), and taking the optimal combination of pre-computed solutions to the smaller problems.

But remember here that our impact estimates for 10-minute intervals tended to grow sublinearly as our added volume increased, at least for the small ranges of volume increases where we had ample data. Also, the approach we took above for combining our 10-minute interval estimates into a 20-minute interval estimate implicitly assumed that there is no natural tendency towards reversion, and that the distribution of market profiles we face in the second interval is independent of our actions in the first, just starting from a new higher price. All of this together means that the "optimal" solution will be to place all of the volume in a single 10-minute interval (at least up to the amount of volume we have in our chart above).

This may be true up to a point, but it may also be an artifact of our model's assumptions. To test and potentially loosen these assumptions, a natural next step is to look at pairs of market profiles for adjacent 10 minute intervals and see if their distributions display meaningful correlations rather than approximate independence. If this is the case, we will need to adjust our calculation of the expected impact in the second interval based on a new distribution of market profiles, informed by our actions in the first interval. We will develop this line of inquiry in our design of Proof's trading algorithm, which will be documented in a separate, forthcoming whitepaper. For now, we will go back to considering single intervals of time, but try to extend on analysis to longer intervals.

7 Modeling for 1 hour intervals

We'll next apply the same approach with a different base unit of time: 1 hr instead of 10 minutes. Things get a little clunky given that 1 hr units don't quite perfectly partition the trading day, which lasts for the 6.5 hours from 9:30 am to 4 pm. To deal with this, we'll just focus on the time period from 10 am to 3 pm each day. Hence, each trading day will correspond to 5 disjoint observation windows per day. (This means we are discarding trading time periods near the auctions.) We'll stick with most of our other choices initially, like rounding UP_v, $DOWN_v$, and $NEUTRAL_v$ to two decimal places. However, it wouldn't make sense to use the same ranges to study our observed Δ_p distributions, as the longer time period is likely to lead to the Δ_p values scattering more diffusely across a wider range of values. Hence, we'll recalculate new ranges using the same approach we did before - approximately breaking our entire set of observed values into ranges each containing a similar number of observations.

As before, we end up with rough characterizations of Δ_p distributions corresponding to each fixed (rounded) combination of values for UP_v, $DOWN_v$, $NEUTRAL_v$. We did this separately for two time periods: January 2019 through April 2019, and May 2019 through August 2019. (We increased the length of each time period from 3 months to 4 months, as a small gesture of compensation for the fact that there are many fewer disjoint 1 hr windows than 10 minute windows, hence our sample sizes are much smaller.) We can now perform the same sanity checks we performed above: first, we'll estimate a weighted "average" of these Δ_p distributions. Next, we'll compute a weighted average statistical distance between our Δ_p distributions and this average. Doing this on the first data set for January through April in 2019, we get approximately 0.35 as the weighted average statistical distance. This indicates that our estimated distributions are considerably distinct from one another.

Next we test the stability of our estimate over time by comparing our estimates from the first data collection period with our estimates from the second data collection period: the weighted average statistical distance between the estimates (matched up by identical UP_v, $DOWN_v$, $NEUTRAL_v$ values) turns out to be around 0.16. This is comparatively high and a bit troubling. One could argue that 0.16 is still significantly less than 0.35, and so there is still evidence of meaningfulness here, but we are stubborn optimists, and we believe better models should exist. Hence we will refuse to be satisfied with this state of affairs, and we will explore further ideas.

So let's take a step back and reconsider the nature of our problem. By changing our fundamental unit of time from 10 minutes to 1 hr, we have effectively cut our sample size by a factor of 6. We should also expect that the distribution of values for UP_v, $DOWN_v$, and $NEUTRAL_v$ will be spread out across a wider range, thereby further decreasing the sample size at any particular combination of these values if we continue rounding them to two decimal point precision. All of this can understandably result in an increase in the expression of noise relative to the expression of meaning in our data.

There are some obvious things we can try to counter this relative amplification of noise. First, we can simply increase the overall length of the time period used for data collection - e.g. collecting data for each of the training and test data sets over 6 month periods. This is not a sustainable strategy for dealing with increasing noise, however, as our data access is limited (we currently have access to less than three full years of historical data as part of our OneTick service) and we would expect that the patterns we are seeking to fit do shift somewhat overtime. Hence, older and older data is likely to provide decreasingly meaningful predictions of future behavior.

Another thing we could try is non-disjoint observation windows. If, for example, we started a new one-hour observation window every 10 minutes, we would have roughly the same number of data points we had when were considering disjoint 10 minute windows. Interpreting our data, however, becomes a bit messier. For instance, things that happen near the beginning or end of the trading day are included in fewer observation windows than things that happen near the middle of the trading day (unless we include more partial windows that really don't morally match our targeted length). Also, as the windows are overlapping, they do not represent independent samples and may display misleading patterns coming from induced correlations. Nonetheless, this may be a viable and helpful approach, and we will likely investigate it in the future.

For now though, we have a more radical idea. When variation in the observations threatens to overwhelm the foundation of a modeling framework, there are two somewhat opposite paths to consider: 1. adding complexity to try to model additional sources of variation and hence remove them from the "noise" terms, or 2. radically simplify the modelling framework. Let's pick path 2. for now.

A good process for simplification is to ground yourself with the question: what am I ultimately trying to predict, and what intermediary steps between my observations and my final prediction can I possibly skip or condense? We are ultimately trying to predict how the distributions of Δ_p values change as a function of increases in UP_v. There is no requirement that we estimate individual distributions of Δ_p corresponding to individual values of UP_v, $DOWN_v$, and $NEUTRAL_v$ in order to do this.

There are a few more direct routes that we could take to our desired destination. As mentioned briefly above, we could try to fit Δ_p as a function of UP_v, $DOWN_v$, $NEUTRAL_v$ by imposing a form for the function. For instance, we might use linear regression to find the best fitting model of the form:

$$\Delta_p \approx c_0 + c_1 UP_v + c_2 DOWN_v + c_3 NEUTRAL_v,$$

where c_0, c_1, c_2, c_3 are fixed numbers chosen to minimize distances between our observed Δ_p values and the linear model outputs. We highly suspect from our study of 10-minute intervals above, however, that a linear model will not be a good fit for the relationship between Δ_p and the values UP_v, $DOWN_v$, and $NEUTRAL_v$. Other functional forms might be promising, however, but it's pretty tricky to guess a good one without exploring the data more first.

Another thing we can do is stick with our approach of matching up observations that have the same (rounded) $DOWN_v$ and $NEUTRAL_v$ values, but not bother to group them by these values. To see more precisely what this means, let's imagine a concrete example. Suppose we have a pair of observations:

$$(\Delta_p^1, UP_v^1, DOWN_v, NEUTRAL_v) = (-0.002, 0.12, 0.05, 0.02),$$

$$(\Delta_p^2, UP_v^2, DOWN_v, NEUTRAL_v) = (0.001, 0.18, 0.05, 0.02)$$

Now, before we were viewing the first as being a sample from the distribution of Δ_p when $(UP_v, DOWN_v, NEUTRAL_v) = (0.12, 0.05, 0.02)$, and the second as being a sample from the distribution of Δ_p when $(UP_v, DOWN_v, NEUTRAL_v) = (0.18, 0.005, 0.02)$. We were trying to estimate all these various distributions independently. Then we were assembling these individual estimates into an overall estimate of how each amount of increase in UP_v affected the distribution of Δ_p.

Examining estimates of distributions of Δ_p conditioned on the values of $(UP_v, DOWN_v, NEUTRAL_v)$ may be interesting in its own right, but really there is no reason to combine individually noisy estimates to estimate a quantity that we could instead estimate directly. An alternative view is to interpret our pair of observations above as a single sample

$$(\Delta_p^2 - \Delta_p^1, UP_v^2 - UP_v^1) = (0.003, 0.06).$$

Here we are sampling from the distribution of $\Delta_p^2 - \Delta_p^1$, conditioned on the $UP_v^2 - UP_v^1$ value. We can throw all such samples together, ignoring differences in the $DOWN_v$ and $NEUTRAL_v$ values, except for the requirement that they need to match in order to generate such a sample of $(\Delta_p^2 - \Delta_p^1, UP_v^2 - UP_v^1)$. Since the number of common values for $UP_v^2 - UP_v^1$ is much smaller than the number of relevant combinations of values for $(UP_v, DOWN_v, NEUTRAL_v)$, this means we will have more data relative to the difficulty of our chosen estimation task.

Let's step through more explicitly how this view compares with our previous approach. Before, the value Δ_p^1 would have gone into the computation of the average Δ_p value for $(UP_v, DOWN_v, NEUTRAL_v) = (0.12, 0.05, 0.02)$, while the value of Δ_p^2 would have gone into the computation of the average Δ_p value for $(UP_v, DOWN_v, NEUTRAL_v) = (0.18, 0.05, 0.02)$. We'll let N_1 denote the total number of observations with $(UP_v, DOWN_v, NEUTRAL_v) = (0.12, 0.05, 0.02)$, and we'll let D_1 denote the sum of the Δ_p values for all of these observations *except* our Δ_p^1 data point. Similarly, we'll let N_2 denote the total number of observations with $(UP_v, DOWN_v, NEUTRAL_v) = (0.18, 0.05, 0.02)$ and D_2 denote the sum of the Δ_p values for all of these observations except our Δ_p^2 data point.

The contribution of Δ_p^1 and Δ_p^2 to our final estimate of the average Δ_p difference when UP_v increases by 0.06 is then proportional to:

$$N_1 \left(\frac{D_2 + \Delta_p^2}{N_2} - \frac{D_1 + \Delta_p^1}{N_1} \right) = N_1 \left(\frac{D_2}{N_2} + \frac{\Delta_p^2}{N_2} - \frac{D_1}{N_1} - \frac{\Delta_p^1}{N_1} \right)$$

Focusing in on the terms that depend on Δ_p^1 and Δ_p^2, we have:

$$\frac{N_1}{N_2} \Delta_p^2 - \Delta_p^1 \tag{1}$$

Keeping this in mind, let's return to the mindset of looking directly at our matched pairs of observations. Now suppose that we estimate the average Δ_p difference when UP_v increases by 0.06 by taking *all* such matched pairs with $UP_v^1 + 0.06 = UP_v^2$ and averaging the $\Delta_p^2 - \Delta_p^1$ values of these pairs. To write this out in mathematical notation, we'll let $d_1 = \Delta_p^1, d_2, \ldots, d_{N_1}$ denote the set of all Δ_p values corresponding to observations with $(UP_v, DOWN_v, NEUTRAL_v) = (0.12, 0.05, 0.02)$ and we'll let $f_1 = \Delta_p^2, f_2, \ldots, f_{N_2}$ denote the set of all Δ_p values corresponding to observations with $(UP_v, DOWN_v, NEUTRAL_v) = (0.18, 0.05, 0.02)$. Then our new computation includes:

$$\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} f_j - d_i$$

The final value after adding in the analogous sums for other matched combinations of $(DOWN_v, NEUTRAL_v)$ values with UP_v differing by 0.06 will be divided by the total number of individual terms $f_j - d_i$ to produce the final average, but we won't worry about that normalization for now. If we scrutinize the above formula carefully, we see that it can be rewritten as:

$$= N_1 \left(\sum_{j=1}^{N_2} f_j \right) - N_2 \left(\sum_{i=1}^{N_1} d_i \right)$$

Recalling that $f_1 = \Delta_p^2$ and $d_1 = \Delta_p^1$ for the particular pair of observations that we have been following, we see that their contribution to this value is:

$$N_1 \Delta_p^2 - N_2 \Delta_p^1. \tag{2}$$

Up to the normalization which we have ignored, this is the same as the contribution we found in (1). To make it the same, we should divide by N_2. Thinking through this another way, we realize that we don't quite want all matched pairs to have equal weight. In keeping with our approach before, we want the total contribution for each observation like Δ_p^1 to be uniform, regardless of how many observations Δ_p^2 it gets paired with. Dividing by N_2 accomplishes this. We are then sticking to the principle that probability mass is inherited from the baseline observations *without* our activity added, and seeing more samples with our activity added can yield more accurate estimates of the resulting distributions, but it doesn't change their relative weight in our analyses.

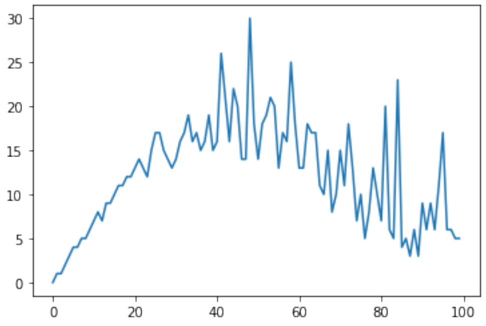
All of this means that we can ultimately arrive at the same kind of estimates we were computing before more directly, without going through estimates of the average Δ_p for each distribution corresponding to fixed values of $UP_v, DOWN_v, NEUTRAL_v$. Of course, if it is the same result, there is no reason it will be any better! But the important takeaway here is that we might not need all of the individual distribution estimates to be robust in order for our final estimate of the impact to be robust. We can instead compute the final estimate directly and perform sanity checks on the final estimate, without going through the individual Δ_p distributions for different $(UP_v, DOWN_v, NEUTRAL_v)$ values as an intermediary step.

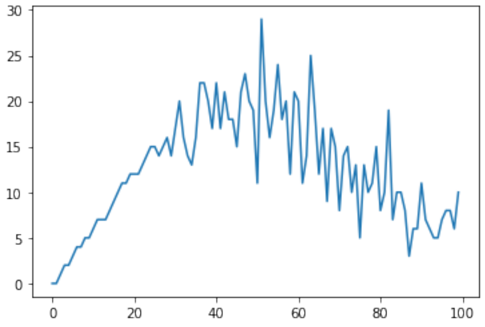
So how might we check the robustness of the final result? And if we're not going to provide estimates of the Δ_p distributions for fixed values of $(UP_v, DOWN_v, NEUTRAL_v)$, then what else might we want our model to provide? If we think of the $f_j - d_i$ values above as samples of a distribution corresponding to the change in Δ_p resulting from an UP_v increase of 0.06, then there are many things we can attempt to measure about this distribution beyond its average. For any quantities we might attempt to measure, one natural sanity check is time stability: if we compute them over two data sets of disjoint time periods, do we get similar results? Of course, there may be meaningful changes in the patterns over time, but time scales of say a few months remain a good candidate for these checks.

In order to get a more holistic sense of the distribution of $\Delta_p^2 - \Delta_p^1$ values over our matched samples, we'll compute percentiles that divide our sampled values into 10 ranges, each range containing an equal number of samples. The middle breakpoint will be the median, and hence graphing the location of this as a function of the change in UP_v gives us a sense of the "average" impact of various increases in UP_v. We can do the same for other breakpoints, and get a sense of how the distributions as a whole respond to increases in UP_v.

We'll do this for two sets of data, a three month period covering January through March 2019, and the next three month period covering April through June 2019. For this calculation, we chose to only match pairs of data points with the same $(DOWN_v, NEUTRAL_v)$ values *and occurring in the same symbol.* In our prior approach for 10-minute intervals, we were implicitly matching such pairs across symbols as well. We suspect that limiting our matches to occurring within symbol might help eliminate some noise in our matched samples, even though it will result in fewer matched samples. We'll combine over symbols by weighting by average daily notional value. In other words, data for symbols that trade greater typical notional values will be weighted proportionally heavier in our results.

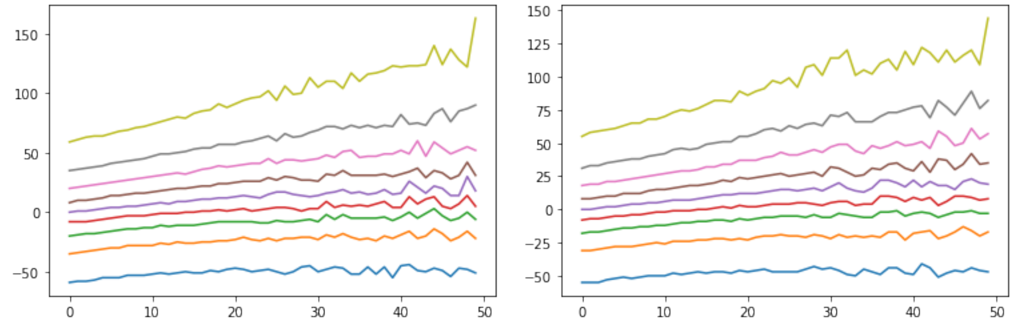
Here are graphs of the medians, with the y-axis showing the median Δ_p value (recall that this is relative price change over the 1 hr time interval), and the x-axis starting from 0 change in UP_v and going up to 10% change in UP_v (i.e. UP_v increasing by +0.1). The first graph is for the earlier time period, and the second graph is for the later time period. The x-axis of additions to UP_v have been multiplied by a factor of 1,000 to make the numbers easier to read. Thus, a value of "1" on the x-axis corresponds to an addition of +0.001 to UP_v (an addition of 0.1% of the ADV of the course of an hour). The y-axis of Δ_p differences has been multiplied by a factor of 10,000 to make the numbers easier to read. Thus, a value of "1" on the y-axis corresponds to a difference of 0.0001 between the Δ_p values of the matched samples.





There are significant differences here of course, but the general pattern is undeniably similar. As we increase UP_v, we see a relatively linear rise in our median estimated impact at first. Then the pattern degenerates to noise, and ultimately is likely swallowed by bias in the market conditions under which atypically large trading deviations occur.

Now let's take a look at all the percentiles graphed similarly together. We'll cut off the increment to UP_v at 5% so we can focus in on the more meaningful part of the graphs:



You may notice that the break points for the extremal case of UP_v's increase being 0 are perfectly symmetric around 0. This is not a coincidence or a bug in our data processing. When the match criterion for comparing pairs of observations is that the UP_v values should be the same, then each $f_j - d_i$ that goes into the calculation is balanced by the value of $d_i - f_j$, which is also thrown in. This creates the symmetry we see in the breakpoints at the far left side of the graph. We can visually identify a few more interesting effects: the highest percentiles we are tracking move much more than the lower ones, and all seem to increase for awhile before devolving into noise. This qualitatively matches the behavior we found for 10-minute intervals previously.

It is encouraging that the patterns of behavior here seem to be stable enough over time for us to obtain qualitatively the same results for these two time periods. This is good evidence that our model is capturing something meaningful. However, there are some important limitations we must keep in mind. Firstly, "meaningful" does not guarantee "meaning what we think it should mean." There are still several questionable assumptions embedded in the combination of our model design and our interpretation of its results. In particular, interpreting these plotted curves as showing us the distribution of the change likely to result in Δ_p from our trading activity still involves assuming that UP_v and only UP_v will move as a result of our actions. We know this assumption is unrealistic, but the key question is *how* unrealistic? Trying to understand and mitigate the error introduced by this assumption is a core question for our continuing research.

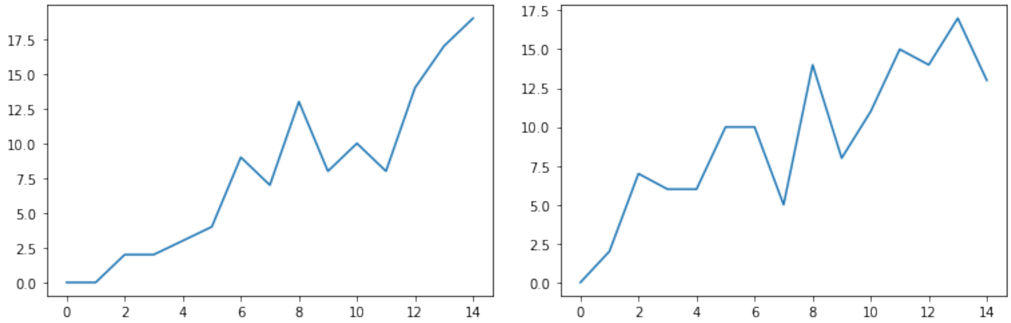
Secondly, by streamlining our model and computing our final estimates of the $\Delta_p^2 - \Delta_p^1$ distributions in this way, we have lost opportunities to customize our model to symbols. We are now directly aggregating our samples of $\Delta_p^2 - \Delta_p^1$ across our symbols into one distribution and empirically computing percentiles. When we were going through the intermediate step of computing estimates of distributions for various $(UP_v, DOWN, NEUTRAL_v)$ values for 10-minute intervals, we had a relatively easy way to combine data across all symbols to compute those estimates, and then customize the weighting of those individual estimates by symbol. This seemed like a reasonable compromise between having no symbol customization and sharding our data into over 8000 disjoint pieces, drastically shrinking our sample sizes. It wasn't clear though, how much customization mattered, even for the 10-minute model above. In order to get a sense of whether this kind of customization is worth it, we should go back and investigate how different the weights end up being as we range over symbols - it's possible they don't actually differ that much on average, and the customization was more flash than substance.

We can also investigate the potential for meaningful customization by dividing stocks in a modest number of meaningful groups: e.g. dividing them into a small number of categories based on average daily volume or notional value. We could then do the same computations separately on each group and compare the results: if they are not discernably different, we may doubt the benefits of customization. If they are discernibly different, we may continue (conservatively) slicing the data a bit further to see if we can find further benefits to even more refined groupings, while still checking our results for robustness and trying to make sure they do not devolve into coincidental quirks of insufficient sample sizes. This is something we will investigate in a later section of this paper.

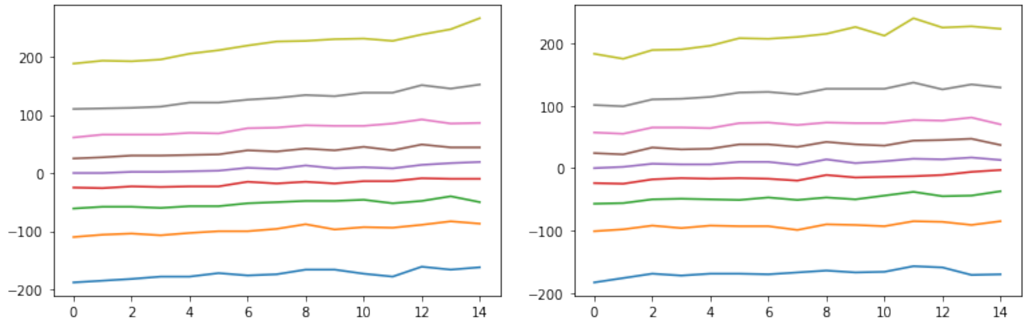
8 Modeling for 1 day intervals

To model time units of 1 trading day, we'll employ the same approach we took in the previous section for modeling 1 hour intervals. The only real difference is that our values of Δ_p are computed over trading days (comparing the closing trade price vs. the opening trade price for the interval from 9:30 am to 4 pm), and our $DOWN_v$ and $NEUTRAL_v$ values will be matched based on their roundings to the nearest 10% rather than the nearest 1% of ADV.

Here are the graphs of the median price impact values as a function of increase in UP_v, first for the earlier time period of January through March 2019, and then for the later time period of April through June 2019. Our x-axis for percentage increase in UP_v will range from 0 to 15% ADV (not far beyond this, the patterns seem to devolve into noise again):



The pattern again seems meaningfully stable, and qualitatively has a similar shape to what we found for shorter time intervals. Now let's take a look at all the percentiles graphed similarly together:



9 Revisiting and tweaking our methodology

At this point, we can look back on what we have done so far as a good proof of concept. Qualitatively robust, meaningful trends in the data do seem to arise from our approach, at

10 minute, 1 hour, and 1 day timescales. But as we prepare to finalize our methodology and implement it in a tool that allows us to visualize the results and make trading decisions, we should revisit a few questionable choices we have made so far.

One key choice was our labeling trades as “UP,” “DOWN,” or “NEUTRAL” based on their prices as compared to the prevailing NBBO. This seems like a reasonable thing to do, but it doesn’t match exactly with the way large institutional orders tend to be traded. When large institutional orders are sliced up into smaller orders, many of the smaller orders trade more passively, executing at the near side or the midpoint of the NBBO, rather than the far side. In our labeling scheme, many of these smaller trades would get marked as “DOWN” or “NEUTRAL,” even if the larger original order is a buy order.

However, it is common (in our experience) for these more passive trades to occur in inopportune moments. For example, an order to buy at the NBBO price might execute just before the NBBO moves downward. These kind of phenomena make labeling based on the prevailing NBBO a little misleading.

To smooth out the influence of NBBO transitions on a millisecond time scale, we can try a slightly different labeling scheme. We can look instead at how the NBBO moves on a slightly longer timescale around the trade. More precisely, we’ll compare the prevailing NBBO at the time of the trade to the prevailing NBBO one second later. If the midpoint of the NBBO has moved down, we’ll label the trade as “DOWN.” If the midpoint of the NBBO has moved up, we’ll label the trade as “UP.” If the midpoint has not moved, we’ll label the trade as “NEUTRAL.” This method of labeling lets us make the market tell us if the trade was part of a moment of downward, upward, or neutral pressure on price. Naturally, this labeling scheme has its own caveats: the NBBO moves in reaction to many things, not just the particular trade, and the 1 second timescale is a bit arbitrary. Nonetheless, we feel that collecting $UP_v, DOWN_v, NEUTRAL_v$ variables according to this new labeling and studying additions to UP_v in this new labeling is a slightly better fit for our purposes.

Another decision worth revisiting is our choice above to measure the *difference* of Δ_p values for each matched pair of observations. This choice makes the interpretation of our output a bit murky and indirect. After all, what we observe is the actual prices and Δ’s between them, so it makes things cleaner if we align our model output with that. This is not too big of a change. For a pair of observations like $(UP_v^1, DOWN_v^1, NEUTRAL_v^1, \Delta^1) = (0.12, 0.05, 0.02, -0.0003)$ and $(UP_v^2, DOWN_v^2, NEUTRAL_v^2, \Delta^2) = (0.18, 0.05, 0.02, +0.0001)$, instead of cataloging this as a sample of UP_v increasing by 0.06 and Δ changing by $\Delta^2 - \Delta^1 = +0.0004$, we’ll consider this to be a sample of UP_v increasing by 0.06 and Δ becoming $\Delta^2 = +0.0001$ as a result. The weight of this sample in our final calculation will be normalized so that all the pairs that match with $(UP_v^1, DOWN_v^1, NEUTRAL_v^1, \Delta^1)$ with UP_v increasing by +0.06 will have equal weight, and the total weight of these samples will be the weight of $(UP_v^1, DOWN_v^1, NEUTRAL_v^1, \Delta^1)$. The philosophy of this is consistent with our initial approach: we view the lower UP_v sample in each pair as being a sample of what typical conditions might be, and we view the higher UP_v sample as what will result when we insert our behavior into those conditions. Hence we should look to the first sample for the relative weight of the pair, and the second sample for the outcome of Δ that we are trying to predict.

We will also reconsider our units for rounding UP_v. For one day timescales, we’ll keep rounding all of $UP_v, DOWN_v, NEUTRAL_v$ to the nearest 1% of average daily volume. But for shorter timescales, a minimal unit of 1% of ADV is quite large. We’d like to support more granular units, but if we universally round everything to the nearest 0.1% of ADV instead, our sample size for each combination of rounded values degrades dramatically. For one hour timescales, we’ll make a compromise. When trying to model the impact of a large buy order (which we expect to add to UP_v), we’ll still round $DOWN_v$ and $NEUTRAL_v$ to the nearest

1% of ADV, but we'll round UP_v values to the nearest 0.1% of ADV. This allows us to model smaller trades over the hour time scale, but we do not suffer from quite as much degradation in sample sizes as a result. We won't worry about the 10-minute time scales. We'll leave those out for now and focus on modeling impact for hour and day long time scales. More research on market impact over 10-minute time scales will be forthcoming in the whitepaper detailing the design of our trading algorithm.

One remaining issue to consider carefully is how we combine information across symbols. There are many choices to be finalized here. First, do we want to match up observations of $(UP_v, DOWN_v, NEUTRAL_v, \Delta)$ across symbols? Initially we did make these matches, but it does seem more natural (and in fact, much faster computationally) to make matches only within a symbol. This of course reduces our sample size relative to making matches across symbols, but we suspect that it might result in higher average quality (a.k.a. less noise) of the matches we do make. So going forward, we'll only match up pairs of observations within the same symbol.

But once we have collected information about pairs of observations within each symbol, how do we want to combine the results across symbols? Not combining them and having over 8000 separate models leaves many of the models with too paltry a sample size to be meaningful. So we want to combine observations across symbols somehow still. One question is: what symbols should be combined together? At the opposite extreme from fitting a separate model per symbol, we could fit a single model across all symbols. But there is also fertile ground in between these extremes. In the next section, we'll discuss ways of classifying symbols into a small number of groups. For each group, we will combine observations across the symbols in the group to obtain a single model. Another question becomes: how do we weight the observations coming from different symbols within the group? A default choice might be to treat them all equally. But going forward, we choose instead to weight the observations for a particular symbol them relative to the average daily notional value traded in that symbol. In this way, symbols that trade more in dollars are treated as more important than symbols that trade less in dollars.

10 Grouping symbols

There are many properties of symbols that we could consider in trying to define meaningful symbol groups. A few obvious ones are average daily volume (ADV), average daily notional value (NV), and volatility. There are many reasonable ways to measure volatility, but since we are focused on distributions of Δ values that represent relative price changes from the beginning to the end of days/hours, we will be interested the volatility of Δ values in particular. For each symbol, we can look over our data set and collect samples of Δ's over days. We can then order all of these samples from most negative to most positive, and we'll take the interquartile range, IQR, (the difference between the 75th percentile sample and the 25th percentile sample) as our measure of how wildly the Δ values tend to vary for that symbol.

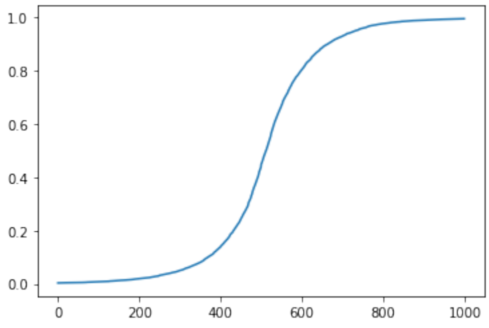
For each symbol, we'll compute three numbers: ADV, NV, and IQR (as defined in the previous paragraph). Are these the only features that might be relevant to the relative price distributions we are studying? Certainly not. But we want to start conservatively here: the more features we use in grouping symbols, the more groups we will have, and more groups means fewer data samples in for each group. In particular, we'll ignore the NV number for now in defining our symbol groups. We will still use it in weighting the samples for different symbols within a group, however.

The values of ADV vary dramatically over symbols. If we want to choose simple orders of magnitude as cutoffs, a reasonable choice seems to be dividing symbols into three categories: those with ADV greater than $100,000$, those with ADV between $10,000$ and $100,000$, and those with ADV below $10,000$. The number of symbols in each group will vary based on what time period we use to compute each IQR, but taking January through March 2019 as an example, we get approximately 3,800 symbols in the high ADV category, 2,900 symbols in the mid ADV category, and 2,200 symbols in the low ADV category.

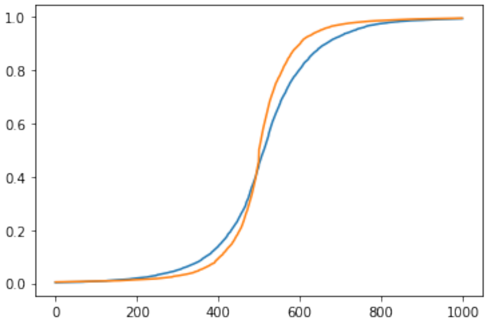
We'll start by considering these three ADV categories as our symbol grouping. For now, we'll set our timescale to be one day. It is easy enough to pair up observations within each symbol according to $UP_v, DOWN_v, NEUTRAL_v$ values and then combine the results across symbols separately within each group, weighting relative to notional value. This gives us three different impact models instead of one: one model for each symbol group. But how do we know if these groupings are meaningful? We can look at how different the three models are, but how do we know if the differences are really caused by distinct behaviors of the symbol groups instead of just random variation in the data? For this, we'll take data over all days from 2019, but we'll divide the data into two pieces, putting every other day into one piece and the remaining days into the other. This gives us two data sets that should behave similarly under any robust model. Now we can train our three symbol group models on each of the two data sets, and we can see if the two versions of each group's models are more similar to each other than they are to the other groups. This is our way of trying to teasing out how much variation is due to noise, and how much variation is really driven by differences between the symbol groups.

To perform this analysis, we need to again choose a metric for comparing probability distributions. We could use the same notion of statistical distance that we discussed in Section 3, but there is one undesirable feature of that metric for our context that we can avoid. That metric compared two probability distributions by measuring how much probability mass had to be moved to turn one distribution into the other. But it did not account for *how far* the mass had to be moved. In our case, we likely want to treat moving some probability mass from say $\Delta = 0.0001$ to $\Delta = 0.0002$ as less meaningful than moving the same probability mass from $\Delta = 0.0001$ to $\Delta = 0.03$. So we will refine our choice of metric here to reflect this.

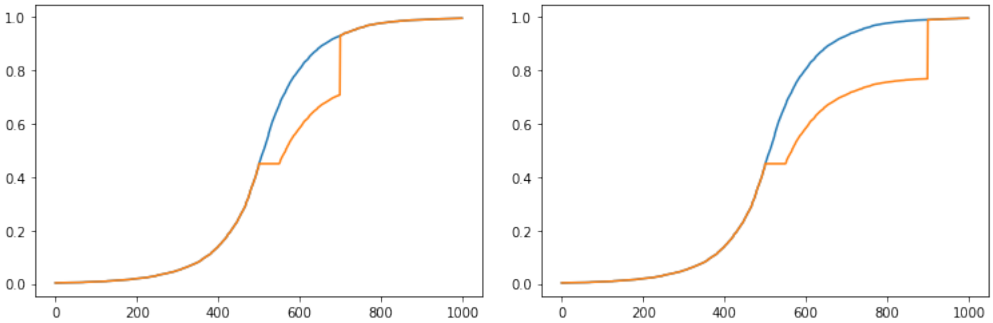
Instead of looking at the individual probability mass on each outcome in isolation, we will instead compute what's called a *cumulative density function*. For each value D that Δ can take, the cumulative density function computes the total amount of probability mass that a distribution places on values $\leq D$. So if we graph the cumulative density function from the lowest Δ value to the highest, we get a curve that goes from 0 to 1 and never decreases. Here is an illustrative example:



If we compute CDFs for two distributions and want to compare them, one option is to compute the area between the curves:



In this example, the blue CDF picks up mass more quickly on the lower values, while the orange CDF picks up more mass on the higher values. The area between the CDF curves can serve as a notion of distance between the distributions that exacts a higher price for shifting probability mass across a wider range of values. We can see this by thinking of a very basic example. If we start with two CDFs that are exactly the same, the area between the curves is 0. (No need to illustrate this, because you won't be able to see the second curve superimposed on top of the first one.) Now let's do something rather inelegant. For one of the distributions, we will take the probability mass accumulated in a certain region and move it all to a single later value. We'll plot the new, altered CDF against the original. Then we'll remove the same mass again, but we'll plop it down on an even higher value:



We moved the same amount of mass, but the resulting area between the curves has grown larger as we extended the wait for the mass to be re-inserted. (If you don't fully trust your eyes to compare the two areas, note that the first carved out shape is actually contained in the second.) You might ask: if this metric is probably better for our purposes than the statistical distance we used before, why didn't we just start with it? Well, we didn't think of it then. We thought of it now. Shrug. (This sort of thing happens all the time. You just don't see it in papers where they write the whole thing pretending they did everything optimally the first time. Fun fact: they did not.)

Let's see what happens when we use this "area between CDF curves" measure to study the various distributions that we built for our three symbol groups on our two data sets. For each group and each data set, we have an estimated probability distribution for each percentage of ADV that we might trade over the course of a day (a distribution of what we expect to happen when we buy 1% of ADV, a distribution for buying 2% of ADV, a distribution for buying 3% of ADV, etc.) Fixing each value of the percentage of ADV that we are modeling, we can compute our metric for how different the two distributions for each group are across our two data sets.

This gives us a sense of how much noise is arising in our modeling process. We can graph the value of this metric as a function of the percentage of ADV we are modeling. Here are the results, for the high, medium, and low ADV symbol groups (in left to right order). The positive integers on the x-axis represent the percentages of ADV that we are modeling buying, and the numbers on the y-axis represent the areas between the two CDFs computed from the two data sets.

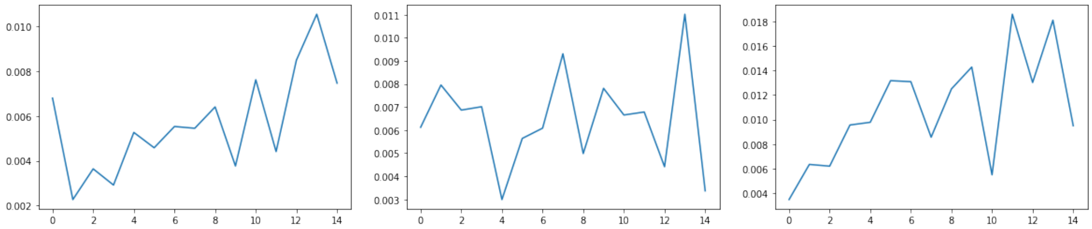


Figure 1: From left to right: High, Mid, and Low ADV groups

We should note a few things here. Firstly, the noise level in our calculations appears to be considerably higher for the low ADV symbol group than the other groups. Secondly, our calculations tend to display more erratic behavior as we model bigger and bigger trades (which makes sense, as the sample size of observations that look like this behavior drops off steeply). Let's compare this to what happens when we compute this same metric across two groups in one of our data sets:

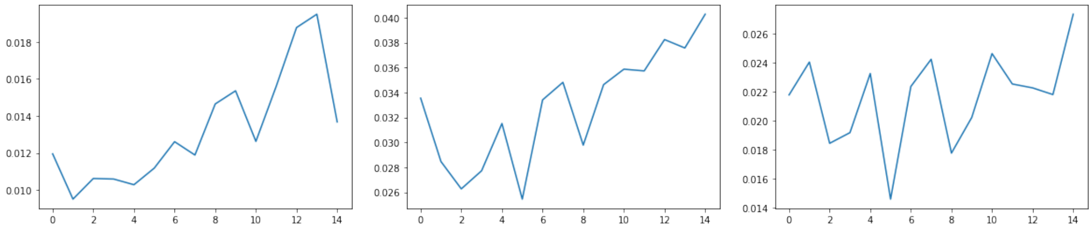


Figure 2: From left to right: High ADV vs. Mid ADV,
High ADV vs. Low ADV, Mid ADV vs. Low ADV

The distances we're getting here between distributions for the Mid ADV and Low ADV groups computed on the same data set are not that much larger than the distances we get between the low ADV group distributions computed on the two data sets. This suggests that modeling this low ADV group separately may not be worthwhile, and hence we'll err on the side of restoring higher sample sizes and re-combine the low and medium ADV groups into a single "non-high" ADV group.

Next let's consider another potentially relevant feature: volatility of daily Δ values as measured by IQR. The values of IQR vary considerably over symbols, mostly within the range from 0.001 to 0.03. (Recall that a IQR of 0.01 for example, means that the 75th percentile of the daily Δ values in this symbol minus the 25th percentiles equals 0.01.) In the interest of choosing memorable round numbers, we can divide the symbols into three categories: those with IQR less than 0.01, those with IQR between 0.01 and 0.02, and those with IQR above 0.02. The number of symbols in each group will vary based on what time period we use to compute each IQR, but taking January through March 2019 as an example, we get approximately 3,400 symbols in the low IQR category, approximately 2,400 symbols in the mid IQR category, and approximately 3,500 symbols in the high IQR category.

If we divide symbols by both criteria (ADV and IQR), we get six groups: high ADV and high IQR, high ADV and mid IQR, high ADV and low IQR, non-high ADV and high IQR, non-high ADV and mid IQR, and non-high ADV and low IQR. Since these names are a little technical and non-intuitive, we'll use the terminology active/inactive for high/non-high ADV, and we'll use stable/volatile/highly volatile for low/mid/high IQR. Our groups will vary in membership depending on what time periods we use to compute ADV and IQR for each symbol, but as an example, using data from January through March 2019, we get the following group sizes: there are 2507 symbols in the stable, inactive group, 1066 symbols in the volatile, inactive group, 1486 symbols in the highly volatile, inactive group, 691 symbols in the stable, active group, 1315 symbols in the volatile, active group, and 1824 symbols in the highly volatile, active group.

Let's take a look at our resulting distributions of price Δ's for some of these groups. To get a sense of what's happening in a particular group on a particular data set, we can plot the 10th, 20th, ..., 90th percentile lines of the Δ distributions as a function of trade size we are modeling. Here are the resulting graphs for our most extreme groups, the stable, inactive group and the highly volatile, active group over one of our data sets:

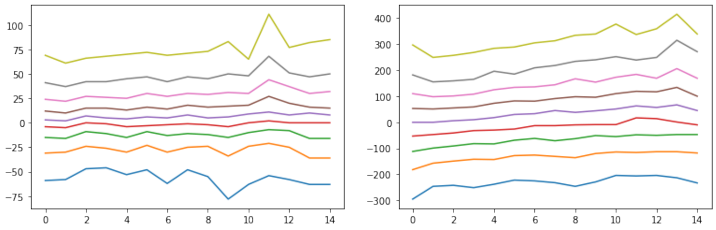


Figure 3: From left to right: stable, inactive group and highly volatile, active group

The units for our x-axis here are $1 = 1\%$ADV for the buying behavior being modeled, and the units for the y-axis are $1 = 0.0001$ for the relative price change Δ.

Here are the analogous graphs for our other data set:

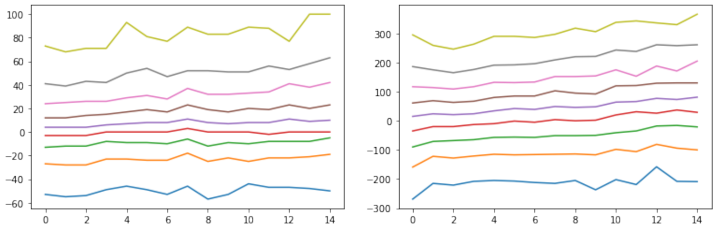


Figure 4: From left to right: stable, inactive group and highly volatile, active group

We can observe here that the these two extremal groups at least behave fairly consistently across the two data sets, and behave distinctly from each other. The greater spread of Δ values for the highly volatile group matches what we would expect from that group by definition, and it's not too surprising that the upward trajectory of the Δ percentiles as a function of the (buy) volume being modeled is cleaner, as we would expect the more active group to have a larger sample size of activity to support the modeling. (This is indeed the case. As a typical example, the distribution for the first data set for $x = 7\%$ADV is based on a sample size of about $5,000$ observations in the stable inactive group, and about $10,000$ observations in the highly volatile active group.)

As above, we can apply our distance metric for comparing CDFs of distributions to get a baseline for how much noise might be in our model outputs per group. Here are the graphs for the six groups, where the distance between empirical distributions is computed across the two data sets, within each group:

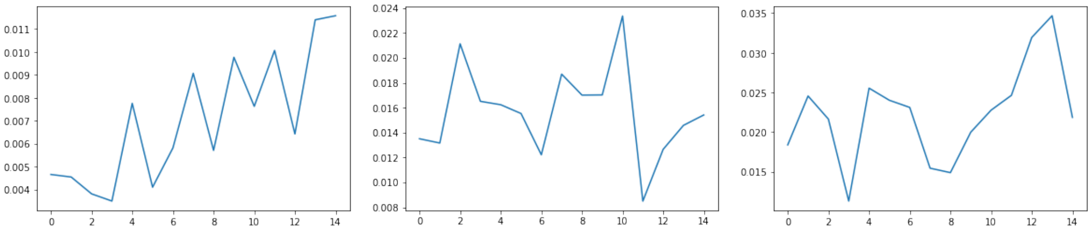


Figure 5: From left to right: stable inactive, volatile inactive, highly volatile inactive

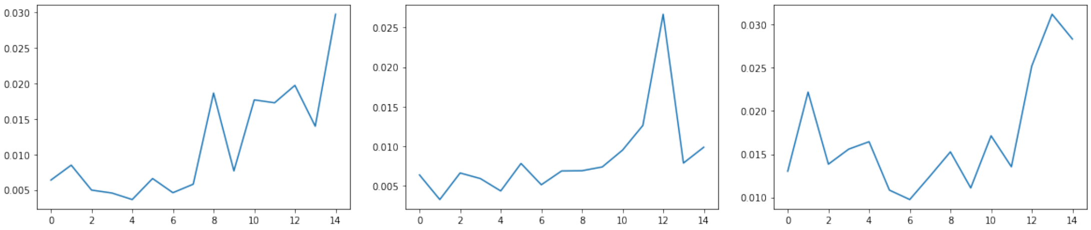


Figure 6: From left to right: stable active, volatile active, highly volatile active

In general, we see there is a trend of increased variation as the volatility increases or the size of the modeled trade increases. Keeping this in mind, let's look now at the levels of variation between some of the groups across the same data set (computed with the same metric and units). We'll start by holding the inactive/active category constant, and we'll look at comparisons across the range of stable/volatile/highly volatile:

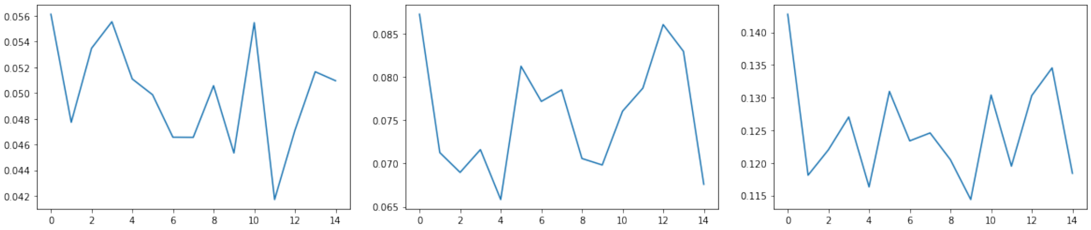


Figure 7: From left to right: (all inactive) stable vs. volatile,
volatile vs. highly volatile, stable vs. highly volatile

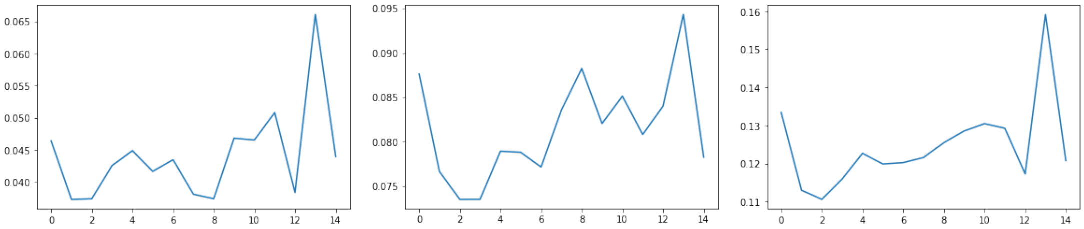


Figure 8: From left to right: (all active) stable vs. volatile,
volatile vs. highly volatile, stable vs. highly volatile

All of these comparisons across groups as the volatility changes show a higher level of difference between the measured distributions than we saw when we held the groups hold constant and varied the underlying data set. This gives us confidence that our grouping by volatility is meaningful.

Now let's look at what happens when we hold IQR category of stable/volatile/highly volatile constant, and vary the inactive/active category:

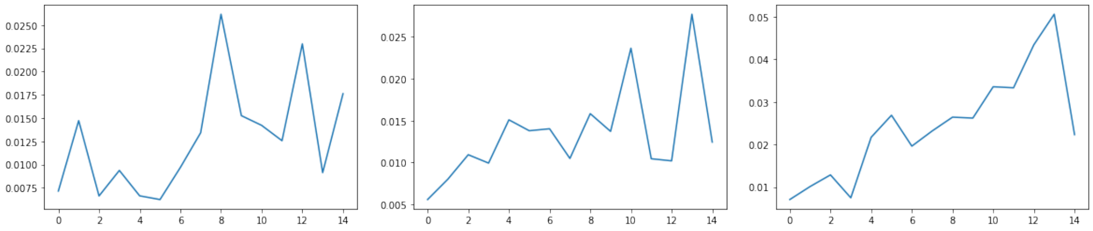


Figure 9: From left to right: (all inactive vs. active) stable, volatile, highly volatile

The comparison here between the stable active group and the stable inactive group on a single data set seems to be similar in magnitude to the comparison of the stable active group to itself across the two data sets. The comparison here between the volatile active group and volatile inactive group is similar in magnitude to the comparison of the volatile inactive group to itself. Only in the comparison of the highly volatile groups does the magnitude here clearly surpass the noise in the individual groups themselves as the trade size increases.

Overall, the evidence for distinct behavior of groups according to volatility categories is much stronger than the evidence for distinct behavior of groups due to activity categories. There is one more trick we can try to make our models a little cleaner. So far we've been computing Δ values from the first and last prices for each symbol, but we haven't tried to account for general movements of the market that affect many symbols at once. Subtracting out these kind of general trends can reduce some of the noise in our Δ values relative to the per-symbol trading dynamics that we are trying to model. We can this "distilling" impact, and we presented our preliminary methodology for it in our previous whitepaper [].

If we apply our distillation technique to subtract out our approximations of wider market effects from the Δ's we are modeling here, we get distributions that are slightly more symmetric and slightly more condensed. For example, here are the 10th, 20th, etc. percentiles graphed as a function of the size of the (buy) trade being modeled, for both the raw and distilled deltas in the highly volatile, active group for one of data sets:

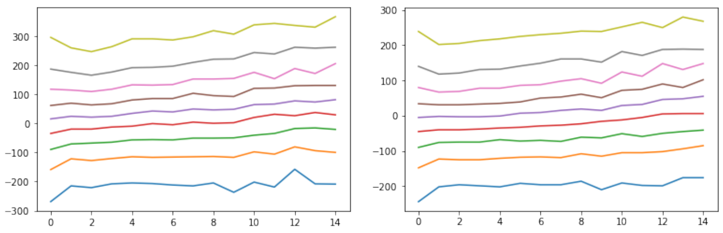


Figure 10: From left to right: raw and distilled distributions for the highly volatile active group

Here are analogous graphs for the stable active group:

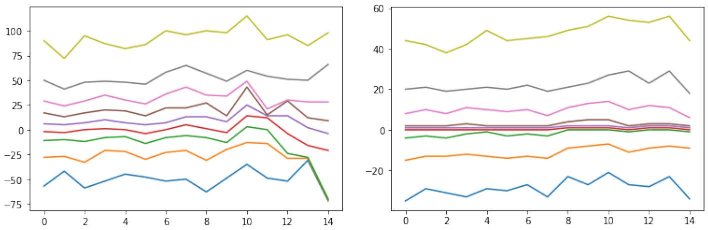


Figure 11: From left to right: raw and distilled distributions for the stable active group

Now let's repeat our analyses using our metric to compare distributions within and across groups. We'll start with the baseline calculation of the groups each compared to themselves over the two sets, but this time with distilled Δ's instead of raw deltas:

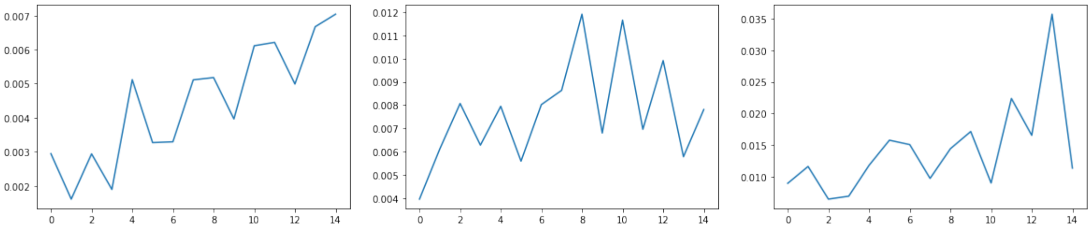


Figure 12: From left to right: stable inactive, volatile inactive, highly volatile inactive (all distilled)

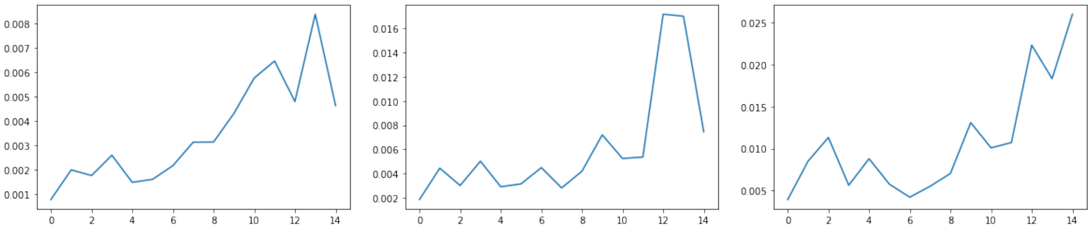


Figure 13: From left to right: stable active, volatile active, highly volatile active (all distilled)

Now let's compare pairs of groups again where the stable/volatile/highly volatile status is held constant, and we vary the inactive/active status:

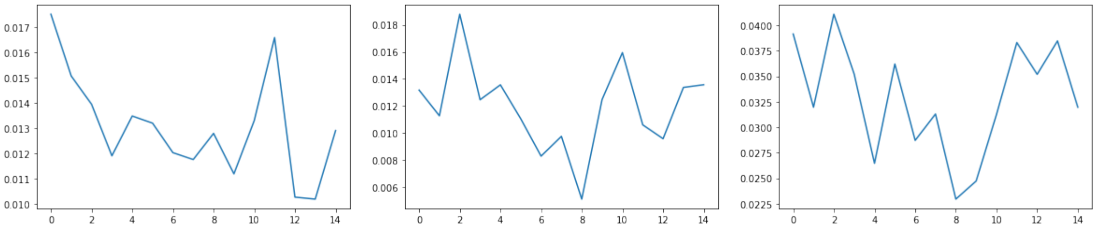


Figure 14: From left to right: (all inactive vs. active) stable, volatile, highly volatile

These graphs suggest that at least for somewhat low values of % ADV being traded, the variation between groups as a function of inactive/active status is a bit higher than the variation within those groups across data sets. As a result, we choose to stick with these six groups of symbols for now rather than condensing them to fewer groups.

11 Summary of our current model and the accompanying pre-trade tool

Putting all of this together, we arrive at what we promised in the beginning: an empirically built model for how the distribution of likely prices changes as a function of how much directional volume we might insert into the market over a specified time scale, customized to symbol groups based on daily price volatility and average daily volume.

We discuss now what kind of questions can we ask such models, and how we might interpret their responses. A question might start with a specific symbol, let's say BAC. Suppose we try to buy a total of 2 million shares of BAC tomorrow. How do we expect our buying activity to affect the price? To get our model to answer to this question, we first have to translate it into the proper units. In recent months (a data set of May through July 2020), BAC has an average daily volume of about 47 million shares, and its daily price fluctuations place it in our volatile, active symbol group. Our proposed size of 2 million is thus roughly 4% of ADV. Our methodology gives us an empirically computed distribution for how we expect the relative price of volatile, active symbols to behave in response to buying 4% of the ADV. We can consult this distribution to extract information like: "there is a 75% chance that the price will increase by at most 1.5% over the course of the day" and "the median price increase under such circumstances is 0.2%." (Note that these numbers are intended to encapsulate the total expected price behavior including market trends, not just the amount "due" to your impact. To gauge how much is "due" to trade impact, you can compare these numbers for different considered trade sizes.) If we want to translate this into dollars instead of percents, we just need to know the starting price of BAC. Naturally, we won't know ahead of time what tomorrow's opening price for BAC will be, but we can get a rough illustration by looking at some other recent price instead. At the moment of this writing for example, the price of BAC is \$25.46. In which case, a median increase of 0.2% would correspond to a price movement from \$25.46 to $1.002 * 25.46 = \$25.51$. We can compare this, for example, to the median increase we expect if we were to try to buy 3 million shares instead, representing about 6% of the ADV. In that case, our data for volatile active symbols suggests a median price increase of 0.3%, corresponding to a price movement from \$25.46 to $1.003 * 25.46 = \$25.54$.

We've provided a pretrade analysis tool that allows you to play around with our data and visualize the distributions we've computed for various trading scenarios. The tool is available at: [link]. It will allow you to graph the distributions that arise from our methodology and get summary statistics. You can choose between seeing raw price deltas and distilled price deltas. You can also choose between viewing price changes in percentages, basis points, or dollars (translation to dollars uses a recent price for the chosen symbol, extracted from a market data source, just like we did in the above paragraph). If distillation is working as intended, distilled price deltas should be more symmetric around zero than raw price deltas, and the distribution should be slightly more concentrated near 0.

As mentioned above, it's important to note that when we are looking at raw price deltas, the general trend of the market over the period of data collection is baked into our numbers. So we should not look at absolute price increases in isolation, but rather should look at how they compare to the expected price change if we trade nothing. In our BAC example, the median price change for a modeled trade size of 0 is +0.1%, suggesting that BAC prices tended to go up over the course of days in our recent data set (which is drawn from the last three months). You can see in the summary stats when we model our buy trade of 4% ADV that the median increase of 0.2% is also reported as (+0.1%) compared to the median we expect when we don't trade. This is intended to give you the proper context of general market activity for this symbol group so you can see the *difference* to expect based on your activity. We could have decided to only show the difference rather than the raw number, but we felt this might create false expectations, as the price you attain is the actual price, not the price adjusted for general market activity. Especially if you later perform analyses on your trades that does not adjust for the general market activity, this could result in a disconnect between context of the model predictions and the context of your analyses. Looking at distilled deltas largely removes this issue.

It's extremely important to keep in mind the severe limitations of our methodology when viewing and interpreting all of these distributions. First, we do not have sufficient sample size to do a separate model for each symbol, so the results are always aggregated over a symbol group, and will never reflect more granular idiosyncracies of a particular symbol.

Second, it's important to remember we do not have a vast historical data set of trades that are labeled by party, so we are matching up pairs of days unlabeled trading days that "roughly differ" by trading activities that "roughly look like" the activity we're trying to model (buying 4% ADV, selling 5% ADV, etc.). Each pair of quotation marks in that sentence hides a multitude of sins. Naturally, as the activity we are trying to model gets bigger (e.g. buying 10% ADV instead of 4% ADV), the number of candidate examples of it that we identify in our historical market data gets smaller. You can see this in the "Sample Size" value that is provided as part of the summary statistics. This is the raw count of how matched pairs of days we found to inform this particular distribution. As the number drops, the quality of the data degrades, and you can see the empirical distributions looking spikier and behaving less intuitively.

On a related note, it's also important to remember that our data matching process looks for the *full* trading activity we are modeling, so this biases our sample to times where that amount of trading was *accomplished* not *attempted*. This bias can be considerable, and certainly grows as the trade size grows. For example, days on which it looks like someone came in and managed to trade 12% of ADV are not only rare, but they are likely to be heavily correlated with market conditions that made it possible for that kind of activity to be accomplished. In some sense, our analysis in its current form is not directly capable of answering the question: "what will happen if I try to buy X shares" but is rather trying to answer the question: "what will happen if I do buy X shares?" The difference here is that the fact of succeeding in completing the trade biases the distribution. Even worse, our approach of matching pairs of trading periods that seem to differ by the activity in question means that for large activities, the trading days that match up as being the same except for missing that large activity are biased to be days with a lower level of activity over all, so that a match with the larger activity present exists in the first place. Days that have no match relative to the activity in question are dropped from the analysis.

Overall, we should apply a quickly increasing skepticism to our numbers as the volume pressure we are modeling increases. But our hope is that for reasonably low volume pressures, our distributions do reflect reality in a meaningful way. Some positive evidence of this is that they behave intuitively for these low pressures, and do seem to be qualitatively robust to our minor modeling decisions. (As detailed above.)

You may notice in the tool that the median of the distilled delta distributions sometimes stays stubbornly put, even as you increase the size of the trade you modeling. This appears to be a feature of the way our distillation methodology behaves. It often places a lot of mass essentially at 0, hence the median stays at 0 for quite some time, even as the overall distribution does begin to shift mass mostly in one direction of the other. This makes the median less useful as a summary statistic for distilled delta distributions.

Addendum When loading very recent data into the tool (specifically data from July-September 2020), we noticed that a few symbols traded remarkably high notional values in this period. In particular, TSLA traded an average daily notional value of nearly 19 billion dollars, dwarfing the median for its highly volatile, active group which was about 7.5 million dollars. Under the notional value weighting within the group employed by our analysis, TSLA represented over 10% of the weight.

Concerned that this would overly skew the analysis and make it less representative for other symbols in the group, we decided to exclue any symbols whose notional value was greater than 1000 times the median notional value of symbols in their group. This filter is pretty mild, and for the time period from July-September 2020, it only excludes three symbols: TSLA, AMZN, and AAPL. (All three of these symbols were in the highly volatile, active group.) We believe this helps protect our group analyses form undue influence of extreme outliers. Since our group analyses are weighted by notional value, this is the same metric we use for our exclusion criterion.

12 Directions for further research

There are several further directions that we are pursuing to enhance this research. They mostly fall into two categories: 1) exploring ways we might make the model simpler and 2) exploring ways we might expand the model's predictive power to a wider and more realistic set of situations. For category 1), we'll continually be looking for ways to simplify our model and adopting any that do not cost us significant accuracy or breadth. For category 2), we'll continually be looking for ways to make our model better fit reality, or at least reasonably fit a wider range of realities. These two categories of improvements should always be pursued in parallel: every time we change our model to make it more accurate or more expansive, we should re-evaluate its complexity and see if the same level of accuracy/breadth/etc. can be achieved in a simpler way. This kind of thinking will protect us from model bloat: a typical consequence of iteratively making models more complex without iteratively simplifying them.

1. Potential simplifications One potential simplification we can explore is a simplification of the market profiles. Do we really need the $NEUTRAL_v$ number, for example? We might hypothesize that this number has less impact on the Δ_p distributions than $DOWN_v$ and UP_v numbers, as volume at the midpoint (hypothetically) does not betray whether the buyer or seller was acting with greater urgency. However, it's also reasonable to guess that the magnitude of the effect of an imbalance between $DOWN_v$ and UP_v might be influenced by $NEUTRAL_v$ value. Another potential simplification could come from trying to combine our models over different timescales (e.g. 1 hour and 1 day) into a single model with time as a parameter. This rests on questions like: what can the various time scale models tell us about each other, and about time scales in between? For instance, how do the models suggest that, say, adding $+1\%ADV$ to UP_v in each hour compares to adding $+6\%ADV$ over a whole day? Can comparisons between our various models guide us in interpreting sequential intervals, which we struggled with applying to our 10-minute intervals so far?

2. Potential extensions and mitigations of model limitations An important next step is to investigate any correlations between the distributions of market profiles for adjacent time intervals. This could potentially allow us to model reversion effects in a realistic way. This will be further explored in our upcoming whitepaper on the design of Proof's trading algorithm.

Another important extension would be to compute not just the expected price impact for longer time intervals when we piece smaller time units together, but also more information about the overall Δ_p distribution. This requires us to think carefully about how to combine distribution models for adjacent time intervals, and also how to designate an “optimal” split of the volume over the two intervals.

Another important direction is to investigate potential effects due to time of day, especially in and around auctions. Right now our 1 hour time intervals are all thrown together to maximize sample size, but this may be mixing legitimately distinct behaviors that appear at different times of day and it would be helpful to separate those out.

Beyond time of day, there may be other helpful features to add to our market profiles or to use to group symbols. Some possibilities include size at the NBBO, relative fraction of trading happen on exchange vs. off exchange, and average trade size. Looking at features of quotes is particularly motivated by considering how a passive-only VWAP algo behaves in practice: buyers trading this way might actually cause the amount of down volume to grow (since they’re just holding the price up), but they’d also probably cause the bid size to be larger than normal and almost certainly cause the bid cancellation rate to be lower than normal. We will pursue these possibilities in our ongoing research, though we must be disciplined about adding features. Adding features will increase model complexity and hence increase the amount of data we need for reliable fitting.

3. Other variations There are many other minor variations to our model that we could investigate, such as looking at volume weighted average prices for each interval instead of last prices, etc. Our main philosophy is to avoid exploring minor variations unless there is a driving reason. As a general rule, testing multiple different minor variations for no particular reason can lead to unnecessary levels of noise in our analysis, and we may become confused as to which changes we’re making are actually having an impact and which are meaningless. That being said, minor variations can be a good source of additional robustness checks, if we are concerned that our results may not be sufficiently vetted by our other sanity checks and we want to confirm that the behavior we observe is qualitatively the same under minor variations of our model definitions.