Coding Notes:
Liquidity Project

December 1, 2024

1 Data

In this document **volume** denotes the number of stocks traded, **value** denotes the product of the price and the number of stocks traded.

1.1 Stocks

We currently use data for the forty stocks that constitute the DAX in the week of January 8, 2024. For these we have 100 levels of the order book.

We see in Figure 1 that 100 order book level cover at least a bandwidth of 120 basis points for bid and 150 basis points for ask. We see in Figure 2 that these 100 levels are at least 8% of the daily trading volume for bids and asks.

Fuer fig 2 besser mit daily min/max des Order books und nicht mit durschnitten! Idee: kumuliere das Orderbook anderes wort als coverage suchen? Order Book Size?

Aktien koennen in anderen DAX Varianten drinnen sind, e.g. MDAX, SDAX, ... There no auction data in several of the files, this is currently being rectified by downloading a fuller dataset for January to June 2024, expected completion of download is September 2, 2024.

1.2 Index

We have data on the DAX (constituents, weights, free float factors) from Eurostoxx since January 2019. The weights of each constituent are plotted in Figure 3 for the period January 2019 to June 2024. The free float of each constituent is plotted in Figure 9 for the period January 2019 to June 2024. Please see the discussion in appendix A.1.

Approximately 50 stocks are part of the DAX at any moment in time between 2019 and 2024. Of these, 24 are continuously part of the index. Two of these, DHL (DPW) and MBG (DAI), changed trading symbol during the sample period.

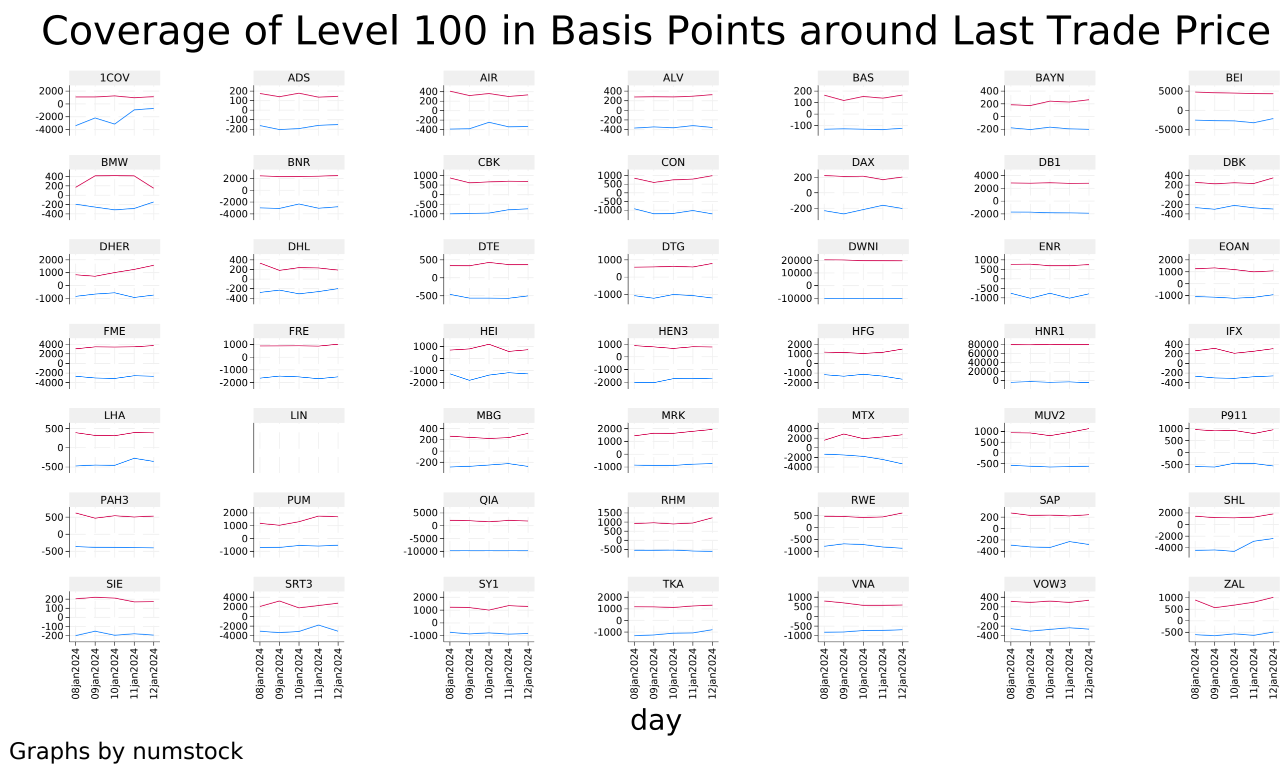


Figure 1: Coverage of Price Bandwidth.

The vertical axis shows level 100 bid/ask prices in basis point deviations from the previous trade price. We see that for all stocks in the sample, 100 order book levels are easily enough to cover the bandwidth vector of {1, 5, 10, 25, 50} that we are currently computing.

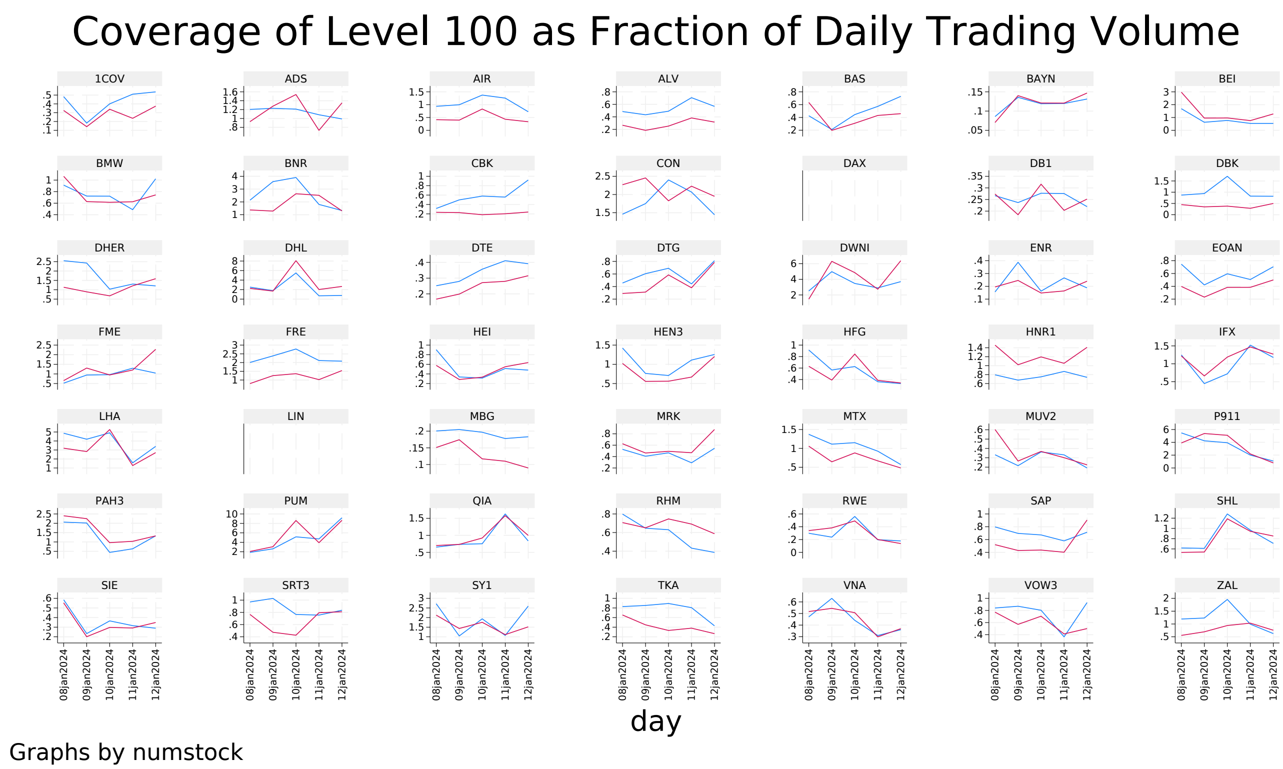
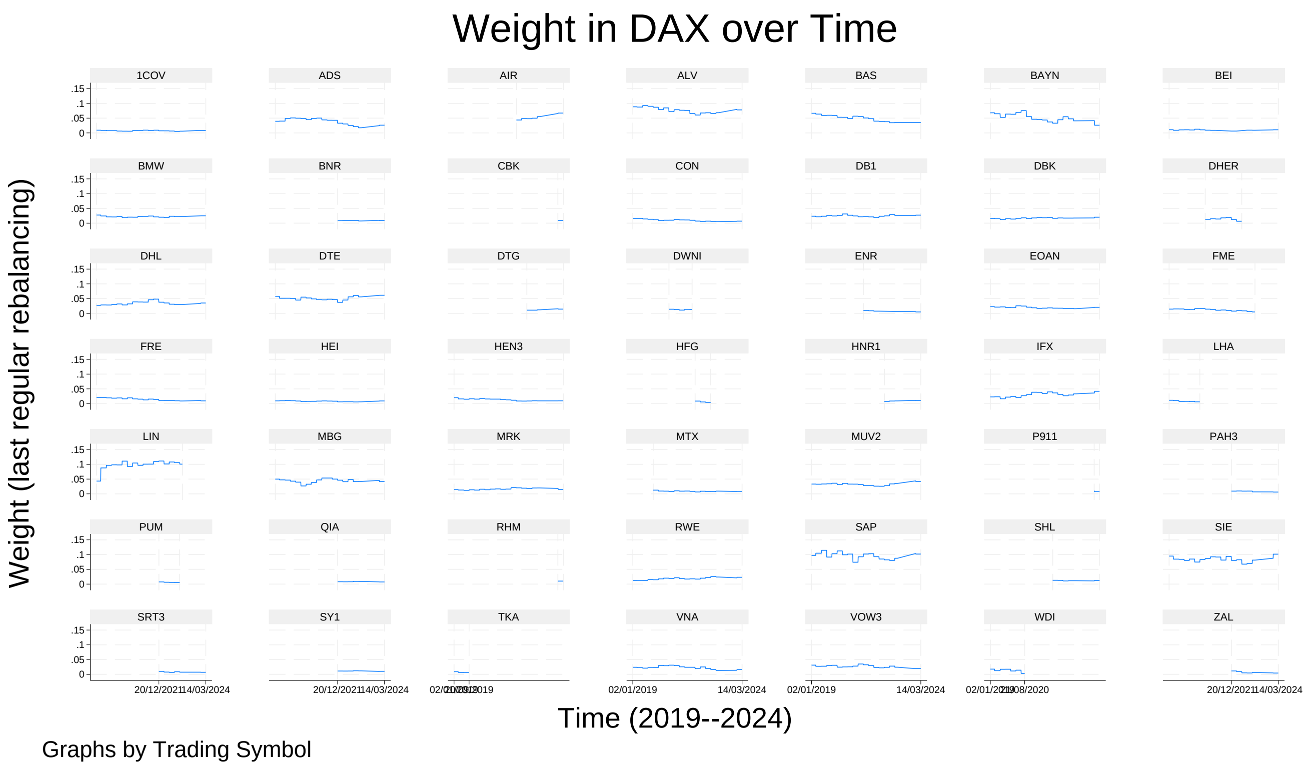


Figure 2: Coverage of Trading Volume.

This data is still without auctions. The vertical axis shows cumulated bid/ask volume up to order book level 100 as fraction of daily trading volume. It does not reach 100% for all the stocks in the sample. **Is this a concern?**



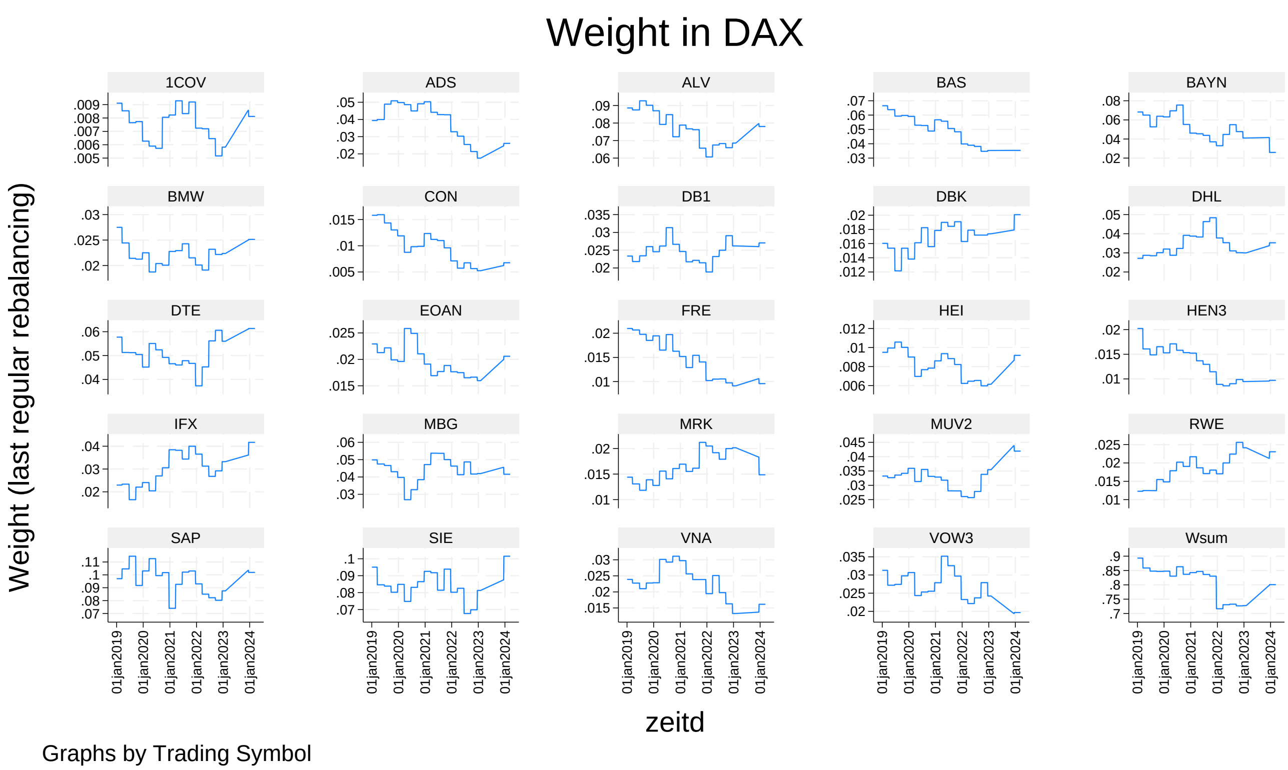
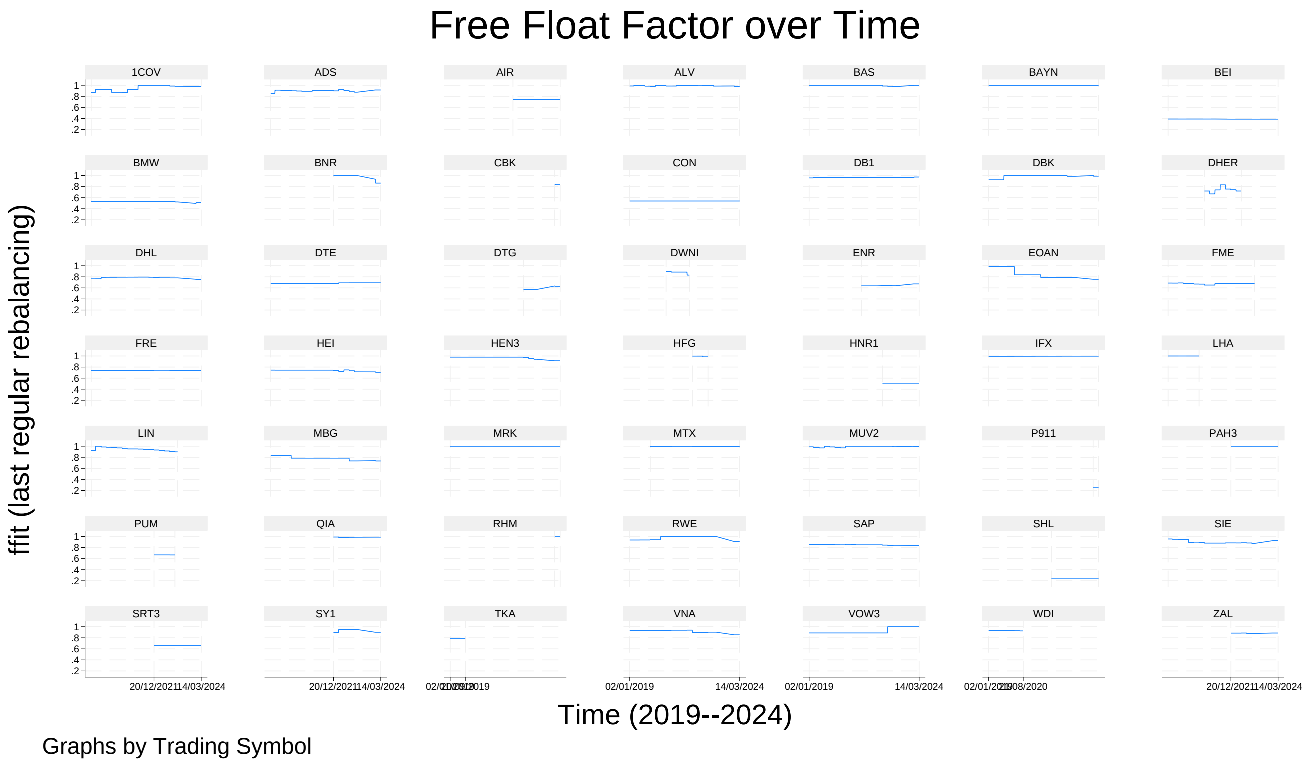


Figure 3: The weights of each stock in the DAX over time.

The top panel shows weights for all stocks that are in the DAX at some time between 2019 and 2024. The bottom panel shows weights only for those stocks that are continously in the DAX for the entire time between 2019 and 2024. DAI and MBG as well as DPW and DHL are treated as one stock, they have identical ISINs. The last graph, labeld Wsum, in the bottom panel shows the sum of the weights for these 24 stocks in the DAX.



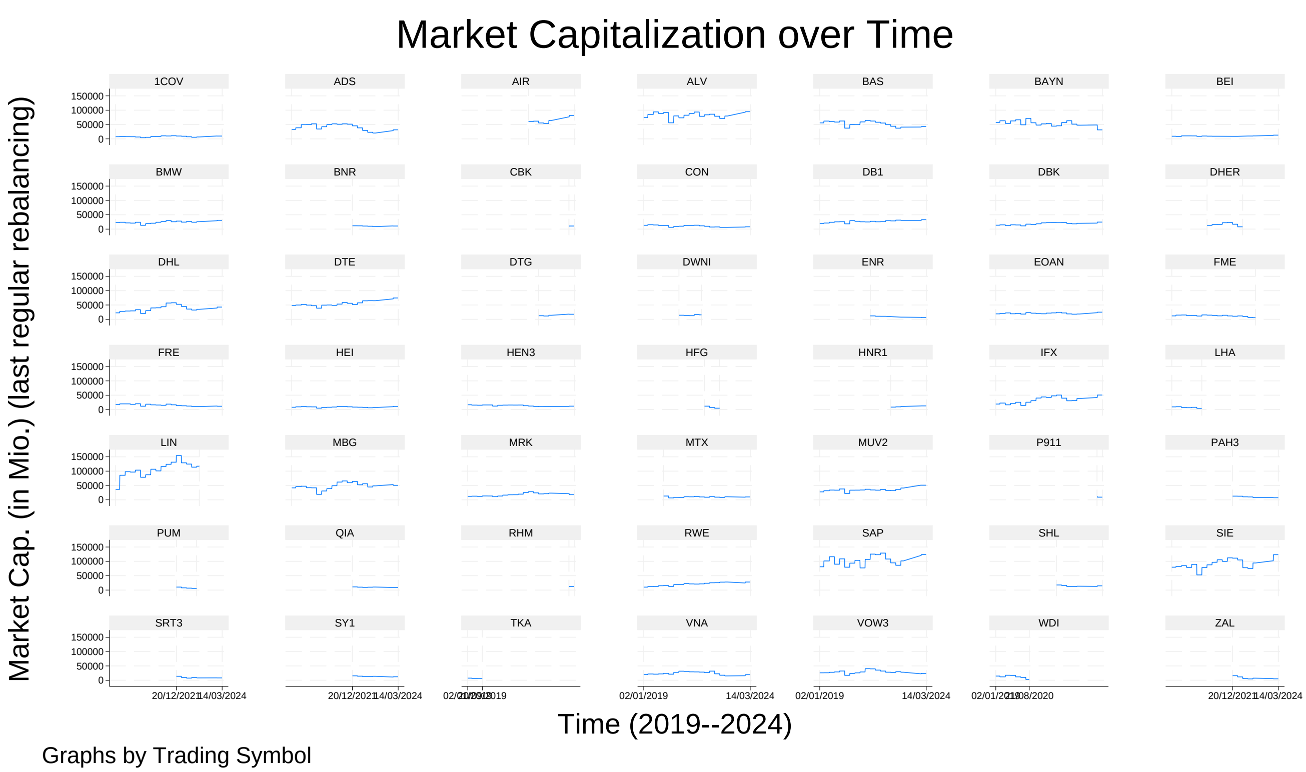


Figure 4: The Free Float Factors and Market Capitalization of each stock in the DAX over time.

2 Formula

Based on Formula 5 of Francioni and Egloff we define for each side of the market S, stock i, and market index M:

$$
\begin{aligned}
{}^{S}\ell_t^i(T,R) &= \sum_{\tau=0}^{n(T)} i + \sum_{\tau=0}^{n(T)} \Delta[I]^S q^{R,ii}{}_{(t_\tau,t_{\tau+1})}, & (1)\\
{}^{S}\ell_t^M(T,R) &= \sum_{\tau=0}^{n(T)} \Delta[I]^S q^{R,iM}{}_{(t_\tau,t_{\tau+1})}, & (2)\\
{}^{S}\lambda_t^i(T,R) &= \frac{{}^{S}\ell_t^i(T,R)}{{}^{S}\ell_t^M(T,R)}\frac{F_t^M}{F_t^i}. & (3)
\end{aligned}
$$

T denotes the length of an interval, for which the statistic is computed, R denotes the bandwidth around the most recent trade price. Francioni and Egloff state that *only increases* in the order book should be considered in computing the sum of $\Delta[I]^S q^{R,i\cdot}{}_{(t_\tau,t_{\tau+1})}$ in equations (1) and(2).

2.1 Computing Changes in the Limit Order Book

Two related questions regarding the computation of $\sum \Delta[I]^S q^{R,i}{}_{(t_\tau,t_{\tau+1})}$ are: How to treat the stock at the beginning of each period T? How to operationalize 'increase'?

Consider the order book as in table 1, where volume level 2 increases to 25, then falls from 25 to 10 and finally increases to 15. Clearly, $\Delta[I]^S q^{R,i}{}_{(t_\tau,t_{\tau+1})}$ is 5 in period 2 and period 4. In period 3 the change in the limit order book is negative.

The final three columns of table 1 illustrate three alternative ways of computing the sum of (positive) changes in the limit order book. Column *inc* takes the formula and text of Francioni and Egloff literally and only sums up the positive increments in the limit order book, leading to a sum of 10. Column *ins* adds the initial stock, leading to a sum of 30. Sidestepping the issue of stocks, increases and decreases of the order book during the period, column *max* takes the maximum length of the order book in the interval, which makes the measure immune to (repeated) insertion and deletion of orders in the order book. Both *inc* and *ins* would continue to rise if a limit order of 5 at price 105 were repeatedly added and then removed again.

Table 1: Order Book Example.

Time	Price 1	Vol 1	Price 2	Vol 2	$\Delta[I]^S q^{R,i}{}_{(t_\tau,t_{\tau+1})}$ inc	ins	max
1	100	10	105	20		20	
2	100	10	105	25	5	5	25
3	100	10	105	10			
4	100	10	105	15	5	5	
					10	30	25

2.2 Computing ℓ for the index portfolio

There is no order book for the index. Instead we compute a virtual order book based on the order book of all constituents. The following example illustrates the process. Consider an index composed of three stocks: A, D, and X, with corresponding weights: $\omega_A = 0.5$, $\omega_D = 0.3$, and $\omega_X = 0.2$. The order book for one one side of the market of each stock and the construction of the virtual index order book at time t are illustrated in table 2.

Let ω_i denote the weight of stock i in the index. Then a simple alternative measure for ${}^{S}\ell_t^M(T,R)$ can be computed as:

$$
{}^{S}\tilde{\ell}_t^M(T,R) \quad = \quad \sum_i \omega_i \quad {}^{S}\ell_t^i(T,R)\,. \tag{4}
$$

The main bottleneck in the computation is the computation of the virtual order book. Working with this alternative measure speeds up the code substantially.

2.3 Computing λ

Analogous to the CAPM, where the β is the regression coefficient when regressing excess stock returns on the market portfolio, we can define $\hat{\lambda}$ as:

$$
\frac{{}^{S}\ell_t^i(T,R)}{F_t^i} = \quad {}^{S}\hat{\lambda}_t^i(T,R)\,\frac{{}^{S}\ell_t^M(T,R)}{F_t^M} + \epsilon_t. \tag{5}
$$

3 Preliminary Results

We show the results for bid and ask, for priceranges of basispoints: $R \in \{1, 5, 10, 25, 50\}$

3.1 Comparison: Virtual Order Book vs. Weighted ${}^{S}\ell_t(T,R)$

See Figure 8

3.2 Comparison: λ vs. $\hat{\lambda}$

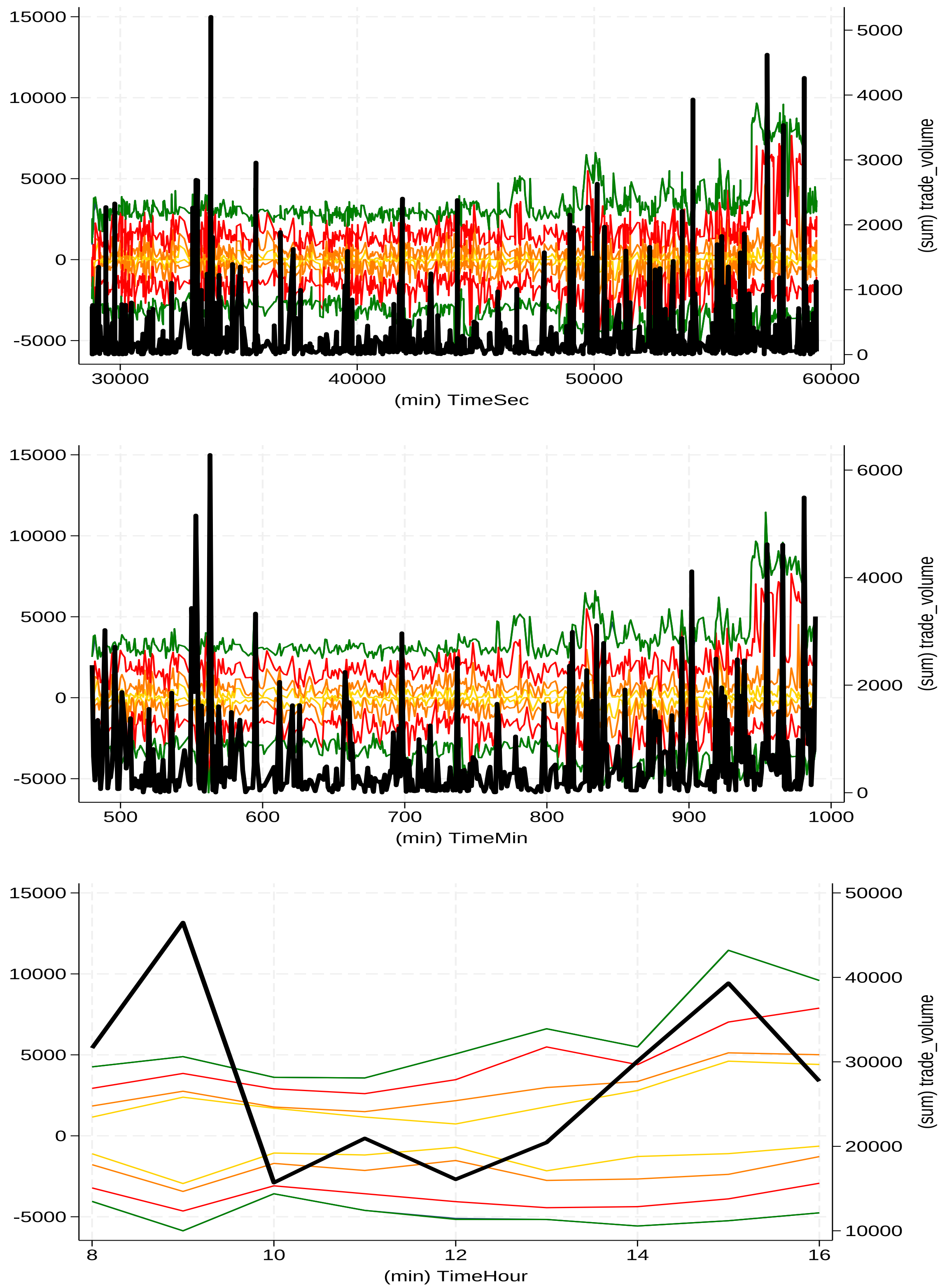


Figure 5: Δ CLOB at Secondly, Minutely, and Hourly Frequency.
This is data for the P911 stock on January 10, 2024. The black solid line represents trading volume. Ask are represented as negative numbers, bid as positive. Same price ranges are represented in same colors. Price ranges are 1, 5, 10, 25, and 50 basis points.

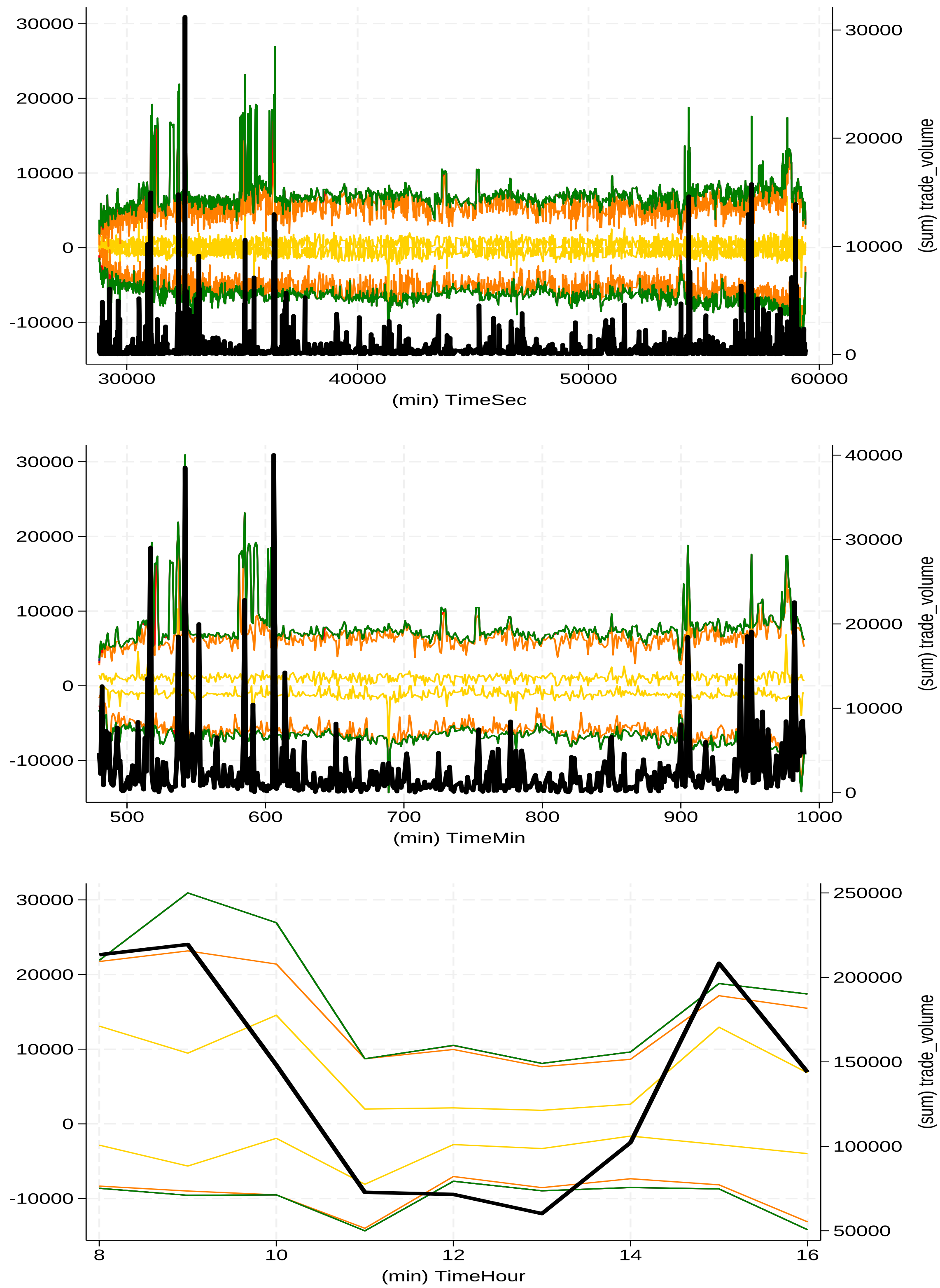


Figure 6: Δ CLOB at Secondly, Minutely, and Hourly Frequency.
This is data for the BASF stock on January 10, 2024. The black solid line represents trading volume. Ask are represented as negative numbers, bid as positive. Same price ranges are represented in same colors. Price ranges are 1, 5, 10, 25, and 50 basis points.

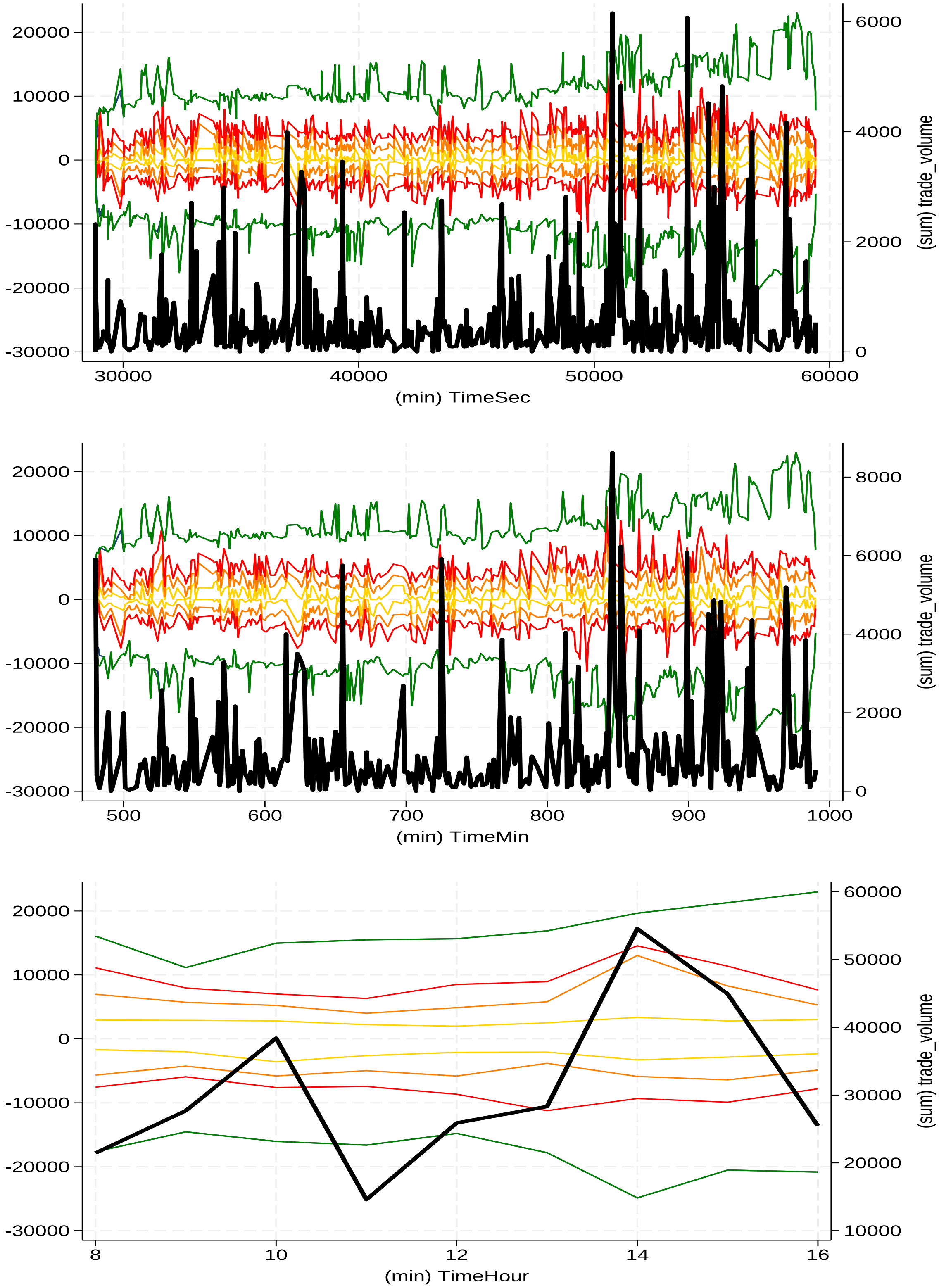


Figure 7: Δ CLOB at Secondly, Minutely, and Hourly Frequency.
This is data for the FRE stock on January 10, 2024. The black solid line represents trading volume. Ask are represented as negative numbers, bid as positive. Same price ranges are represented in same colors. Price ranges are 1, 5, 10, 25, and 50 basis points.

Table 2: Constructing the Virtual Order Book.

Panel A: Order books of individual stocks:

	Stock A		Stock D		Stock X	
Level	Price	Volume	Price	Volume	Price	Volume
1	2	10	3	15	4	15
2	3	20	4	30	6	4
3	4	12	5	3	8	40

Panel B: Reweighted Order books of individual stocks

	Stock A		Stock D		Stock X	
Level	Price	Volume	Price	Volume	Price	Volume
1	1	20	0.9	50	0.8	75
2	1.5	40	1.2	100	1.2	20
3	2	24	1.5	10	1.6	200
Sum		84		160		295

Panel C: Virtual Index Order Book

Level	Price	Volume
1	2.7	20
2	3.2	30
3	3.5	10
4	4	15
5	4.4	9
Sum		84

For this example, index weights are $\omega_A = 0.5, \omega_D = 0.3$ and, $\omega_X = 0.2$. Panel A shows at one moment in time, what price/volume orders are available. Panel B reweights using the index weight of each stock, thus indicating the volume and price for each of the three index components available. To construct the virtual order book in Panel C from Panel B, find the minimum volume available, i.e. the 20 units of A. So the first level of the index order book is 20 units of the index at prices $p_A + p_D + p_X = 1 + 0.9 + 0.8$ which gives the first row of Panel C. For Stock A we have used up the entire available volume of Level 1, there are 40 units in level 2. For Stock D there are still $50 - 20 = 30$ units available, for Stock D $75 - 20 = 55$ units. So the next stack is 30 virtual index units from level 2 for stock A and level 1 for stocks D and X: $p_A + p_D + p_X = 1.5 + 0.9 + 0.8$ which gives the second row of Panel C. This has exhausted level 1 for stock D. There are still 10 units of stock A in level 2, 100 units of stock D in level 2, and 25 units of stock X in level 1. So the next level for the virtual order book is 10 units from level 2 for stock A, level 2 for stock D, and level 1 for stock X. Continue until the first of the stock volumes is completely exhausted, which will be stock A after 84 virtual index units.

Figure 8: Correlations for Virtual Order Book λ vs Weighted ${}^{S}\ell_t\,(T,R)$.
Box Plot of correlation coefficients between the $\lambda(P,T)$ computed via virtual order book and by weighting individual stock ℓ.

A Data Appendix

A.1 Free Float

The source reports free float as a fraction (fff_t), in addition to market capitalization (mc_t), price (p_t) and quantity (q_t)of stocks at time t. Therefore the number of stocks floating freely (F_t) can be computed in two ways:

$$F_t{}^q = q_t * fff_t \tag{6}$$

$$F_t{}^p = \frac{mc_t}{p_t} \tag{7}$$

Most of the time the two approaches yield rather similar numbers. Figure 9 shows the computed series and their differences for all stocks in the DAX portfolio 2019–2024. We chose to go with $F_t{}^q$ because marketcap is reported in millions and there is thus a higher potential for rounding errors in $F_t{}^p$. Furthermore, $F_t{}^p$ is computed by dividing the rarely adjusted market cap by the daily varying price, creating a much more variable measure of the free float number.

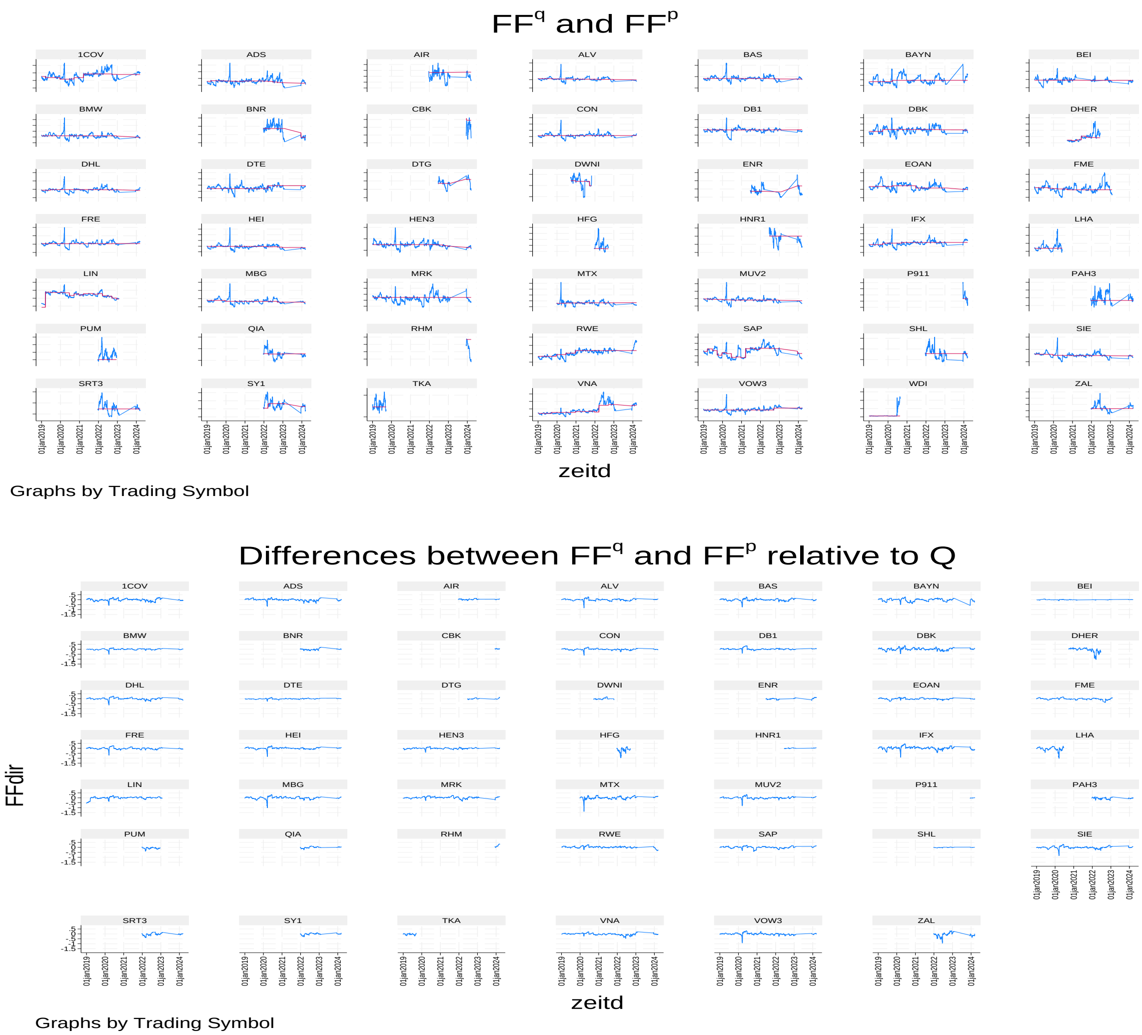


Figure 9: Free Float computed two different ways and the difference for all stocks 2019–2024. The top panel shows the number of freely floating shares computed via quantity and price. ${F_t}^q$ is depicted in red, ${F_t}^p$ in blue. The bottom panel shows $\frac{{F_t}^q - {F_t}^p}{Q}$ for all stocks except WDI on the same scale. WDI is excluded for visual reasons.

B To Do List

1. Page 22, Step 1 of Francioni and Egloff talks about trade volume as number multiplied by price
2. Page 23, Steps 2 and 3 *divide* by the free float. Here again it is important to define clearly. I understand from the text, that F_t stands for the NUMBER of shares floating freely. Most data on free float I could obtain is in percentages, which I call free float factor, fff_t.
3. There are three issues with DAX component stocks:
   (a) DPW and DHL are two different trading symbols connected to the *same* ISIN: DE0005552004.
   I am treating them as the same stock.
   (b) DAI and MBG are two different trading symbols connected to the *same* ISIN: DE0007100000.
   I am treating them as the same stock.
   (c) QIA is associated to two different ISINs: NL0012169213 and NL0015001WM6.
4. For several stocks June 6, 2024 is empty, what happened there?
5. Pick inc/ins/max for the computation of $\Delta CLOB$ (cf. subsection 2.1).
6. Pick virtual order book or weighted sum of stocks for computation of index ℓ (cf. subsection 2.2).
7. Compute λ as ratio of stock and index ℓ or as regression coefficient (cf. subsection 2.3).

C Done

D Time

The timestamp is in Nanoseconds. Stata usually stores data in milliseconds. I carry around:

- timestamp as string
- zeit_Sec: Time of day in seconds
- zeit_Nano: Nanoseconds as extracted from timestamp

so by combining the date, zeit_Sec and zeit_Nano we can perfectly measure time.

E Computing Details for Christian

- clob_ask_vol_Px is the sum of all ask_vol within the band x.
- last_clob_ask_vol_Px is just the previous row of clob_ask_vol_Px .
- max_clob_ask_vol_Px is the maximum of clob_ask_vol_Px between two trades
- delta_clob_ask_vol_Px is the change, i.e. clob_ask_vol_Px - last_clob_ask_vol_Px
- delta_pos_clob_ask_vol_Px is only increases, i.e. max(0,clob_ask_vol_Px - last_clob_ask_vol_Px)