Question:
Can we predict future price movements in financial markets by analyzing prior price and volume data via data science techniques?

More specifically, we will be looking at intraday data for NSE Nifty index futures, and trying to predict returns (in percentage terms, but also as binary / categorical variables) on several time frames. We have price and volume data, and we will construct features from these.

Raw Data:
The dataset is 1-minute open-high-low-close-volume data downloaded from Bloomberg. The data run from August 3, 2015 to February 1, 2016, inclusive.

The NSE Nifty futures trade from 11:45 Singapore time to 18:00 Singapore time. Each row of data represents 1 minute of trading, and there is typically a row of data reported after 18:00, which represents the daily settlement price.

The original fields downloaded from Bloomberg are Date, OPEN, HIGH, LOW, LAST_PRICE, NUMBER_TICKS, and VOLUME.
OPEN: the first traded price in the one-minute period
HIGH: the highest traded price in the one-minute period
LOW: the lowest traded price in the one-minute period
LAST_PRICE: the last traded price in the one-minute period
It is not clear what NUMBER_TICKS represents.
VOLUME: the number of contracts traded in the one-minute period. Do note that the exchange considers the minimum trade to be 75 lots, but Bloomberg reports this as 1. So, if the VOLUME field in our data shows 100, that means that in that minute, we would have seen trades totaling 7500 lots.

As this is a futures contract, there is a monthly roll. i.e., trading is typically most active in the nearest to expiry contract, and then when that expires, trading activity moves to the next contract. As an example, on August 3, 2015, trading was concentrated in the August 2015 contract “NZQ5” (“Q” is the code for August). Then on August 28, 2015, trading moved to the September 2015 contract “NZU5” (“U” is the code for September), because the last trading day of the August 2015 contract was August 27, 2015. Bloomberg takes care of adjusting the reported prices to make a continuous contract, by adjusting the earlier prices. So, the data would be consistent if we wanted to form longer term indicators (e.g., constructing moving averages over more than 1 futures contract cycle such as a 60-day moving average). But, we will focus on short-term (less than 1-day) predictions, and will using indicators that do not look further back than the current day, so the futures contract roll will not have an impact on our analysis even if it was incorrectly done.

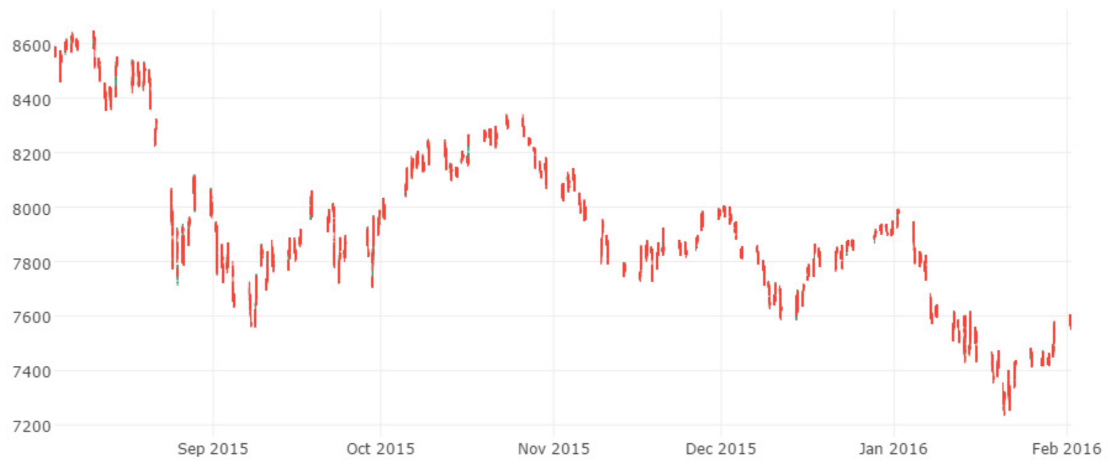


Figure 1 - OHLC chart

In our dataset, the Nifty price has ranged between approximately 7000 – 9000, as shown in Figure 1. Note that the tick size of this contract is very small, at 0.05. Thus, the traded price typically fluctuates through many ticks, and is only very rarely unchanged within a one-minute bar.

| | OPEN | HIGH | LOW | LAST_PRICE | NUMBER_TICKS | VOLUME |
|---|---|---|---|---|---|---|
| Count | 45,609 | 45,609 | 45,609 | 45,609 | 45,609 | 45,609 |
| Mean | 7939.33 | 7941.39 | 7937.24 | 7939.31 | 43.12 | 904.53 |
| Std | 303.85 | 303.72 | 303.99 | 303.84 | 10.96 | 2008.37 |
| Min | 7237.50 | 7241.30 | 7234.55 | 7238.20 | 1 | - |
| 0.25 | 7773.75 | 7775.45 | 7771.65 | 7773.70 | 36 | 239 |
| 0.5 | 7888.60 | 7890.95 | 7886.00 | 7888.45 | 45 | 501 |
| 0.75 | 8136.05 | 8138.00 | 8134.40 | 8136.00 | 52 | 1,049 |
| Max | 8646.00 | 8647.60 | 8643.90 | 8647.00 | 64 | 319,857 |

Table 1 - Summary of Bloomberg data

Table 1 suggests that there may be outliers in the VOLUME data; it does suggest all other fields seem sensible.

| quantile | VOLUME |
|---|---|
| 0.75 | 1049 |
| 0.9 | 2023 |
| 0.95 | 2971.6 |
| 0.975 | 4132 |
| 0.99 | 6154.96 |
| 0.999 | 13213.26 |
| 0.9999 | 32182.85 |

Table 2 - VOLUME percentiles

| Date | |
|---|---|
| 8/25/2015 14:34 | 319857 |
| 8/25/2015 15:07 | 60292 |
| 9/29/2015 13:30 | 49179 |
| 8/24/2015 11:45 | 47967 |
| 8/21/2015 11:45 | 36381 |
| 8/17/2015 11:52 | 28895 |
| 9/9/2015 11:45 | 27515 |
| 8/4/2015 15:12 | 24862 |
| 8/28/2015 11:45 | 24682 |
| 8/4/2015 16:17 | 23441 |

Table 3 - Top 10 VOLUME values

Table 2 and Table 3 together suggest that there is an outlier in VOLUME, we'll drop the highest value of 319,857 as implausible and more likely to be a data error.

Nonetheless, VOLUME takes very high and very low values, such that it only appears normally distributed (with extra weight at volume 0) on a log-log plot:

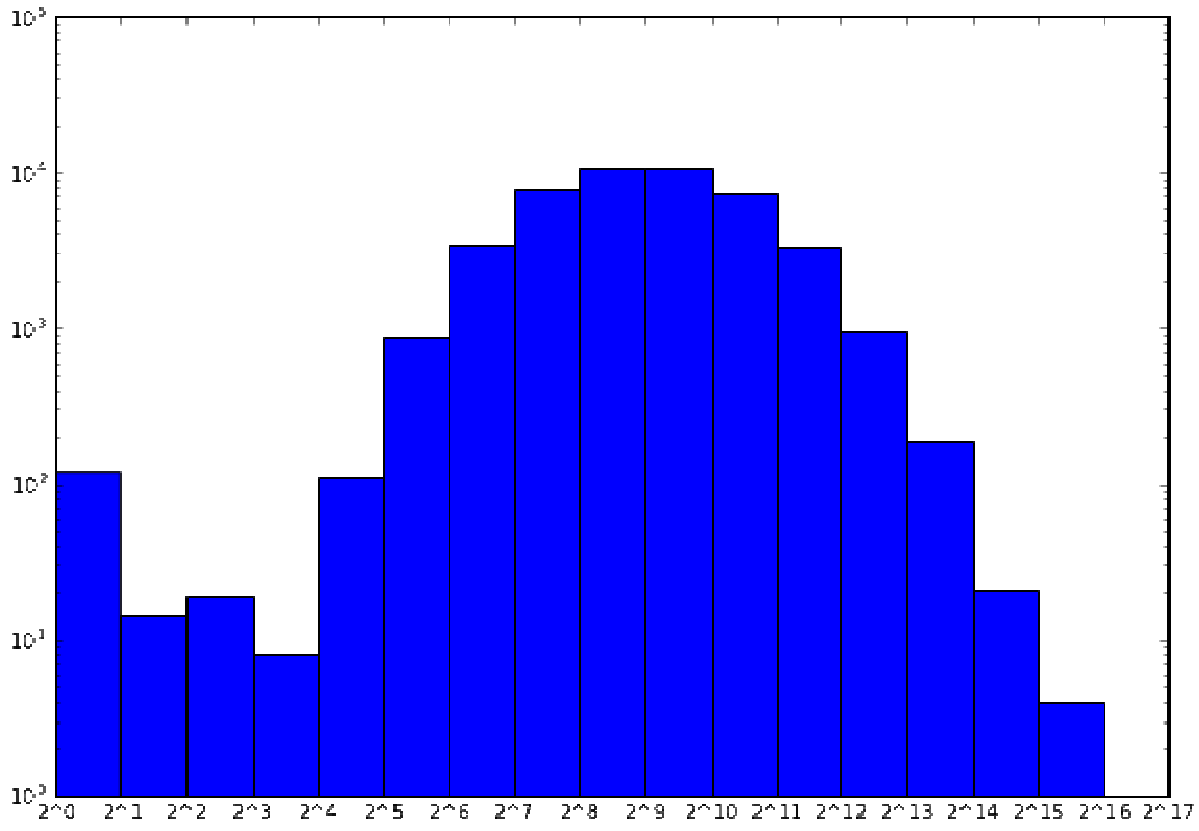


Figure 2- log-log histogram of VOLUME

We may want to consider scaling volume in some cases.

Processing the data

First, we remove the single row where volume was > 300k, as that is likely an outlier. We'll also remove all rows after 1800 (exclusive) as we do not want settlement data – these are not tradable times and these prices are not relevant. We do want the close row (at exactly 1800), because that closing price is important for computing returns on holding periods extending beyond the end of the trading day.

We'll add:
DayOfWeek: 0 for Monday, …, 4 for Friday
Hour: the hour (Singapore time), which ranges from 11 to 18 (but there are few rows at 11 as the market opens at 1145.
We set a time-series index to our dataframe, which will allow us to use 'resample' to create some features on a minute-by-minute basis.

***Response variables***

We'd like to explore predictability on multiple intraday time frames. We'll standardize this list to [1, 5, 10, 20, 40, 80]. i.e., as responses, we create the 1-minute ahead percentage return, but also the 5-minute, …, and 80-minute ahead percentage returns. This will be, for any given row, the (LAST_PRICE x-minutes ahead) / (current LAST_PRICE) – 1. In cases where the x-minute-ahead period extends beyond the end of the trading session, we'll use the LAST_PRICE at 1800 to compute the return. (This is accomplished by 'ffill' in the resample method.)

Table 4 and Figure 3 show the x-minute ahead returns, these appear sensible.

| | 1_min_return | 5_min_return | 10_min_return | 20_min_return | 40_min_return | 80_min_return |
|---|---|---|---|---|---|---|
| count | 45341 | 45338 | 45333 | 45324 | 45305 | 45265 |
| mean | -0.0002% | -0.0011% | -0.0022% | -0.0042% | -0.0068% | -0.0153% |
| std | 0.0412% | 0.0898% | 0.1241% | 0.1730% | 0.2438% | 0.3411% |
| min | -0.5672% | -0.9208% | -1.1510% | -1.4103% | -1.7253% | -2.1843% |
| 25% | -0.0208% | -0.0464% | -0.0633% | -0.0860% | -0.1179% | -0.1681% |
| 50% | 0.0000% | -0.0006% | -0.0006% | -0.0013% | -0.0020% | -0.0094% |
| 75% | 0.0208% | 0.0453% | 0.0611% | 0.0815% | 0.1092% | 0.1459% |
| max | 1.3354% | 1.4400% | 1.5734% | 1.6781% | 1.3665% | 1.8271% |

Table 4 - x-min return summary

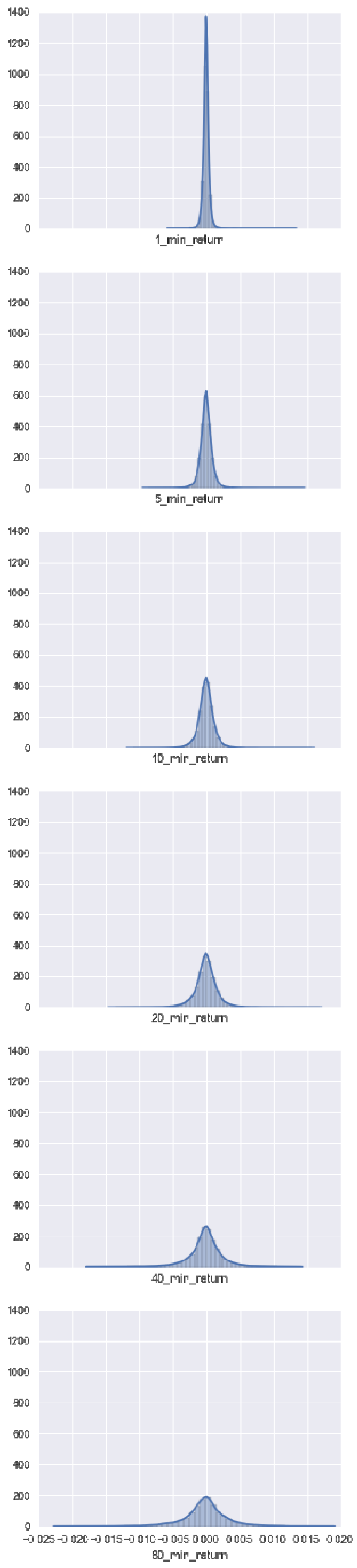


Figure 3 - x-min ahead returns

We look at skewness, kurtosis, and normality of the returns, with the following results:

```
We are using Fisher kurtosis, so a sample from a  normal distribution should
report a 0 kurtosis on this measure
For 1_min_return, skewness is 0.548331; kurtosis is 30.846228
For 5_min_return, skewness is 0.000061; kurtosis is 10.517373
For 10_min_return, skewness is -0.138187; kurtosis is 7.741777
For 20_min_return, skewness is -0.158083; kurtosis is 5.673287
For 40_min_return, skewness is -0.205899; kurtosis is 4.020208
For 80_min_return, skewness is -0.257549; kurtosis is 3.373104

For 1_min_return, p-value of rejecting hypothesis that sample is from a norma
l distribution is 0.000000.
For 5_min_return, p-value of rejecting hypothesis that sample is from a norma
l distribution is 0.000000.
For 10_min_return, p-value of rejecting hypothesis that sample is from a norm
al distribution is 0.000000.
For 20_min_return, p-value of rejecting hypothesis that sample is from a norm
al distribution is 0.000000.
For 40_min_return, p-value of rejecting hypothesis that sample is from a norm
al distribution is 0.000000.
For 80_min_return, p-value of rejecting hypothesis that sample is from a norm
al distribution is 0.000000.
```
Figure 4 - distributional tests on x-minute returns

i.e., the x-minute return data are heavy tailed and thus not normal.

We'll also create 'x_min_updown' columns which are coded +1 if the x-min return was positive, -1 if it was negative, and is blank if it was exactly 0. This ignores very few returns:

```
1_min_updown has 1124 (2.48%) missing values
5_min_updown has 538 (1.19%) missing values
10_min_updown has 453 (1.00%) missing values
20_min_updown has 379 (0.84%) missing values
40_min_updown has 356 (0.79%) missing values
80_min_updown has 351 (0.78%) missing values
```
Figure 5 - missing values in x-minute binary variables

| | 1_min_updown | 5_min_updown | 10_min_updown | 20_min_updown | 40_min_updown | 80_min_updown |
|---|---|---|---|---|---|---|
| count | 44344 | 44945 | 45031 | 45104 | 45131 | 45289 |
| mean | 0.500113 | 0.494872 | 0.495858 | 0.493814 | 0.492455 | 0.482523 |
| std | 0.500006 | 0.499979 | 0.499988 | 0.499967 | 0.499949 | 0.4997 |
| min | 0 | 0 | 0 | 0 | 0 | 0 |
| 25% | 0 | 0 | 0 | 0 | 0 | 0 |
| 50% | 1 | 0 | 0 | 0 | 0 | 0 |
| 75% | 1 | 1 | 1 | 1 | 1 | 1 |
| max | 1 | 1 | 1 | 1 | 1 | 1 |

Table 5 - x-minute binary variables summary

Table 5 shows no obvious issues with the binary variables.

To get a sense of the baseline, we look at the percent of returns that are up:

```
For 1 min returns: +ve: 22176; -ve: 22167; i.e., 50.01% up
For 5 min returns: +ve: 22235; -ve: 22694; i.e., 49.49% up
For 10 min returns: +ve: 22319; -ve: 22695; i.e., 49.58% up
For 20 min returns: +ve: 22265; -ve: 22823; i.e., 49.38% up
For 40 min returns: +ve: 22220; -ve: 22891; i.e., 49.26% up
For 80 min returns: +ve: 21765; -ve: 23351; i.e., 48.24% up
```

**Figure 6 - percent positive binary returns**

So, the returns are almost 50% up

We also make categorical variables for the return, where we code the data as ‘2’ if it’s a significantly positive return, ‘1’ if it’s a small return, and ‘0’ if a significantly negative return,. The cut-offs for these classifications are based on percentiles of absolute values, such that approximately 1/3 of the data are ‘2’, 1/3 are ‘1’, and 1/3 are ‘0’. Figure 7 shows the distribution of the categorical variables, there do not seem to be any obvious issues.

```
For 1 min returns: +1: 15169; 0: 15114; -1: 15057
For 5 min returns: +1: 14994; 0: 15113; -1: 15230
For 10 min returns: +1: 14943; 0: 15111; -1: 15278
For 20 min returns: +1: 14876; 0: 15108; -1: 15339
For 40 min returns: +1: 14794; 0: 15102; -1: 15408
For 80 min returns: +1: 14302; 0: 15088; -1: 15874
```

**Figure 7 - count of categorical return variables**

***Feature variables***

Our first features are the x-minute prior returns, and their binary versions. i.e., for a given minute, we compare the current LAST_PRICE to the LAST_PRICE x minutes ago, and compute the percent change. That is x_min_prior. The binary version is x_min_prior_updown, and simply is 1 if x_min_prior is positive, 0 if it was negative, and missing otherwise.

| | 1_min_prior | 5_min_prior | 10_min_prior | 20_min_prior | 40_min_prior | 80_min_prior |
|---|---|---|---|---|---|---|
| count | 45466 | 45462 | 45457 | 45447 | 45427 | 45387 |
| mean | -0.000002 | -0.000009 | -0.000018 | -0.000036 | -0.000059 | -0.000083 |
| std | 0.000411 | 0.000904 | 0.001257 | 0.001757 | 0.00247 | 0.00341 |
| min | -0.005705 | -0.009293 | -0.011644 | -0.014304 | -0.017556 | -0.022331 |
| 0.25 | -0.000207 | -0.000466 | -0.000638 | -0.000873 | -0.001207 | -0.001685 |
| 0.5 | 0 | 0 | 0 | -0.000006 | -0.000007 | -0.000076 |
| 0.75 | 0.000206 | 0.000459 | 0.000625 | 0.000848 | 0.001168 | 0.00161 |
| max | 0.013178 | 0.014195 | 0.01549 | 0.016504 | 0.013481 | 0.017943 |

**Table 6 - x-minute prior return summary**

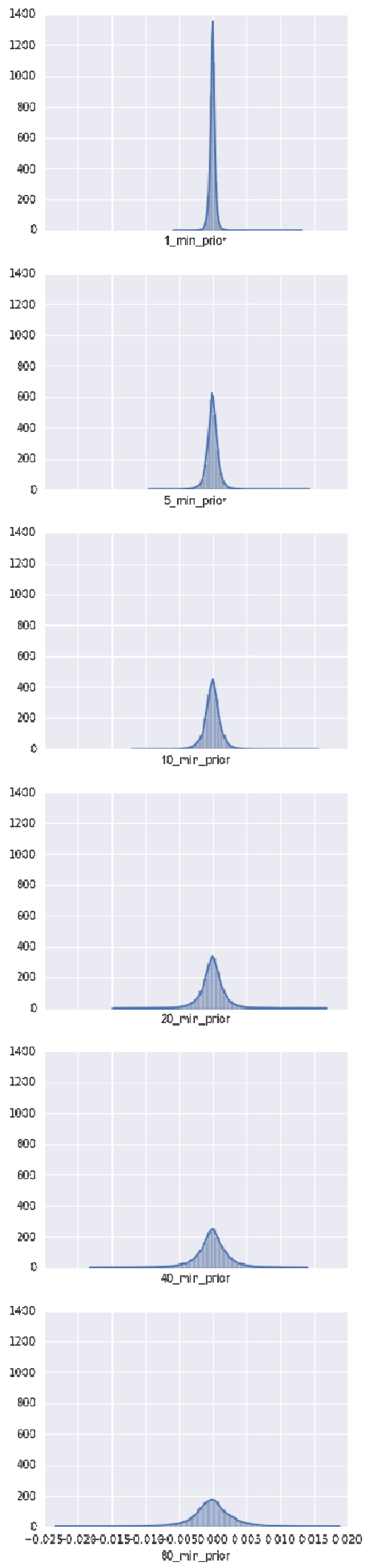


Figure 8 - x-minute prior distribution

Table 6 and Figure 8 show the prior returns, there are no red flags here.

We next make x-minute moving average columns. So, this is the average LAST_PRICE over the last x minutes, including the current minute (because, we assume you would put on a trade at the LAST_PRICE). When we are at times near the start of the day, and there are < x minutes available, we just average over all minutes since the open. So, if it is 1148, the 5-minute moving average would only average over the LAST_PRICE for 1145, 1146, 1147, and 1148. Table 7 summarizes these columns – there seems to be no cause for concern. Figure 9 shows a sample of these series.

| | 1_min_ma | 5_min_ma | 10_min_ma | 20_min_ma | 40_min_ma | 80_min_ma |
|---|---|---|---|---|---|---|
| count | 45467 | 45467 | 45467 | 45467 | 45467 | 45467 |
| mean | 7939.34 | 7939.38 | 7939.42 | 7939.52 | 7939.67 | 7939.86 |
| std | 303.86 | 303.85 | 303.85 | 303.84 | 303.83 | 303.85 |
| min | 7238.20 | 7241.47 | 7242.00 | 7247.19 | 7252.31 | 7264.39 |
| 25% | 7773.78 | 7773.52 | 7773.52 | 7773.31 | 7773.77 | 7773.96 |
| 50% | 7888.60 | 7888.72 | 7888.74 | 7888.64 | 7889.38 | 7890.20 |
| 75% | 8136.00 | 8136.17 | 8135.83 | 8135.95 | 8136.53 | 8139.01 |
| max | 8647.00 | 8644.35 | 8643.02 | 8642.07 | 8641.23 | 8639.23 |

Table 7 - x-minute moving average summary

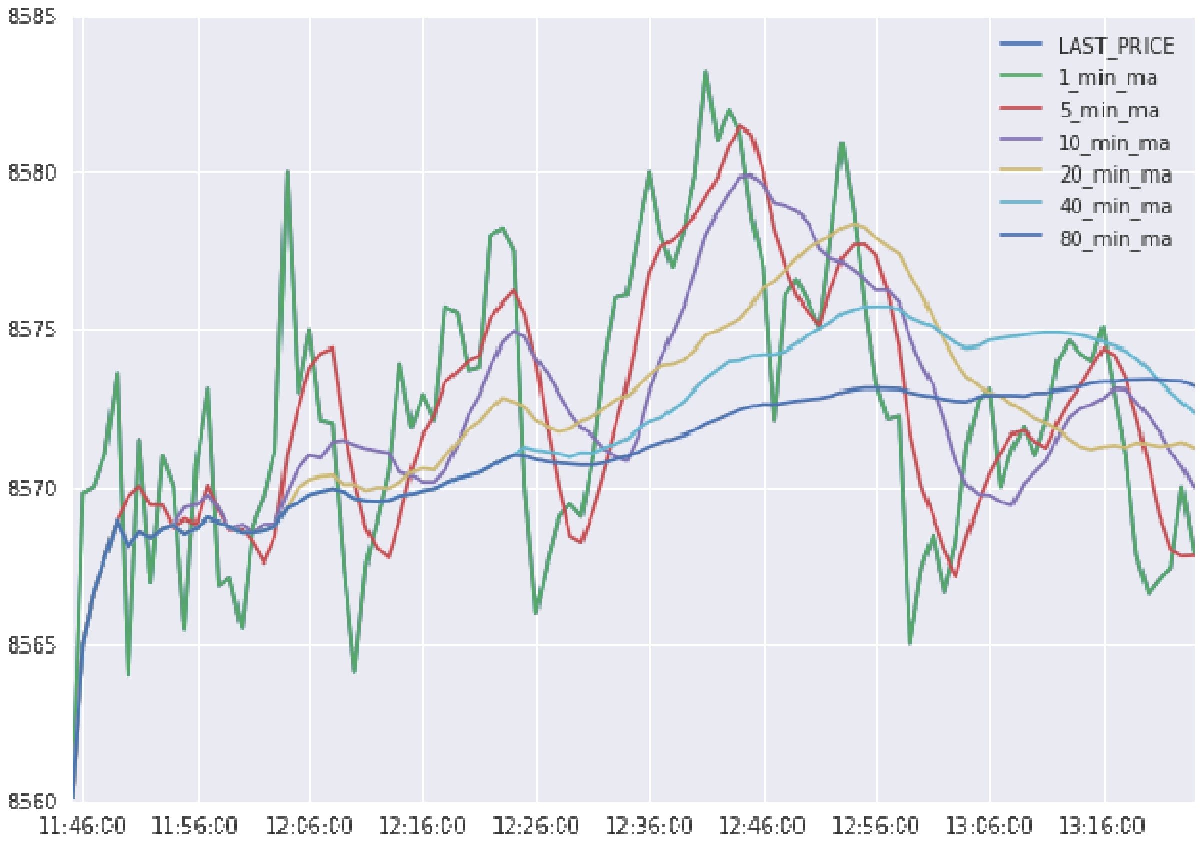


Figure 9 - example of moving averages

We also create x-minute difference from moving average columns, and binary ("updown") versions of these.

|  | 1_last_vs_ma | 5_last_vs_ma | 10_last_vs_ma | 20_last_vs_ma | 40_last_vs_ma | 80_last_vs_ma |
|---|---|---|---|---|---|---|
| count | 45,467 | 45,467 | 45,467 | 45,467 | 45,467 | 45,467 |
| mean | 0.00 | -0.03 | -0.08 | -0.17 | -0.33 | -0.51 |
| std | 0.00 | 3.51 | 5.30 | 7.63 | 10.87 | 15.18 |
| min | 0.00 | -51.77 | -59.05 | -74.44 | -92.14 | -122.13 |
| 25% | 0.00 | -1.75 | -2.66 | -3.74 | -5.18 | -7.11 |
| 50% | 0.00 | 0.00 | -0.03 | -0.04 | -0.05 | -0.25 |
| 75% | 0.00 | 1.73 | 2.61 | 3.59 | 4.89 | 6.64 |
| max | 0.00 | 82.23 | 94.20 | 104.04 | 112.01 | 109.29 |

**Table 8 - price vs moving average summary**

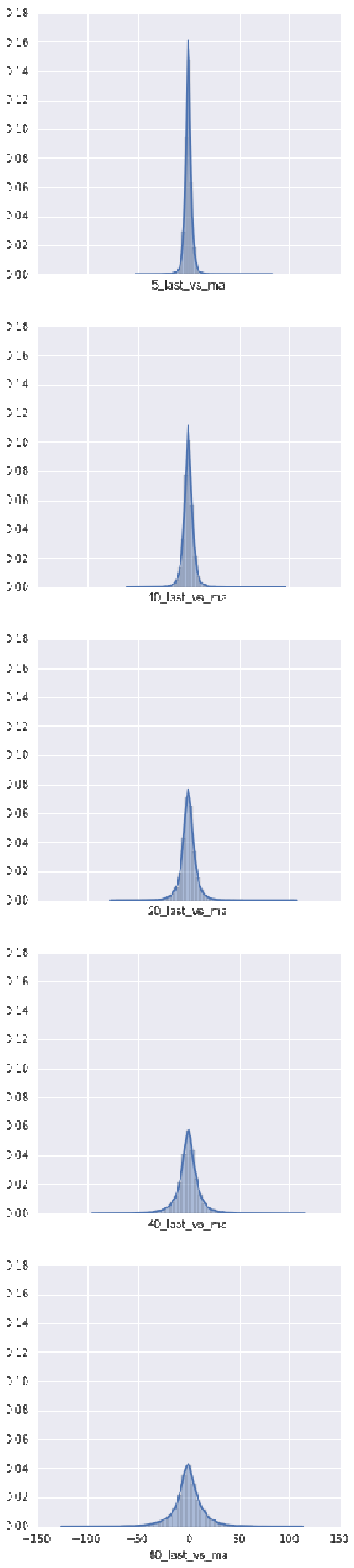


Figure 10 - last price vs moving average distributions

Table 8 summarizes the difference from moving average columns, Figure 10 plots their distributions. This looks fine; we shouldn't use the 1-minute columns (this is just the price minus itself).

We add some features for volume data. These are the rolling 1, 5, …, 80 minute average volume columns. We'll also add some columns where we average the log of (volume + 1) over these periods. We need to add 1 to make sure we get defined numbers in cases where volume = 0. We do these averages by taking rolling sum over rolling count. So, at the beginning of the day, when less than x minutes have elapsed, the x minute average will just be the average over the number of minutes elapsed since the start of the day.

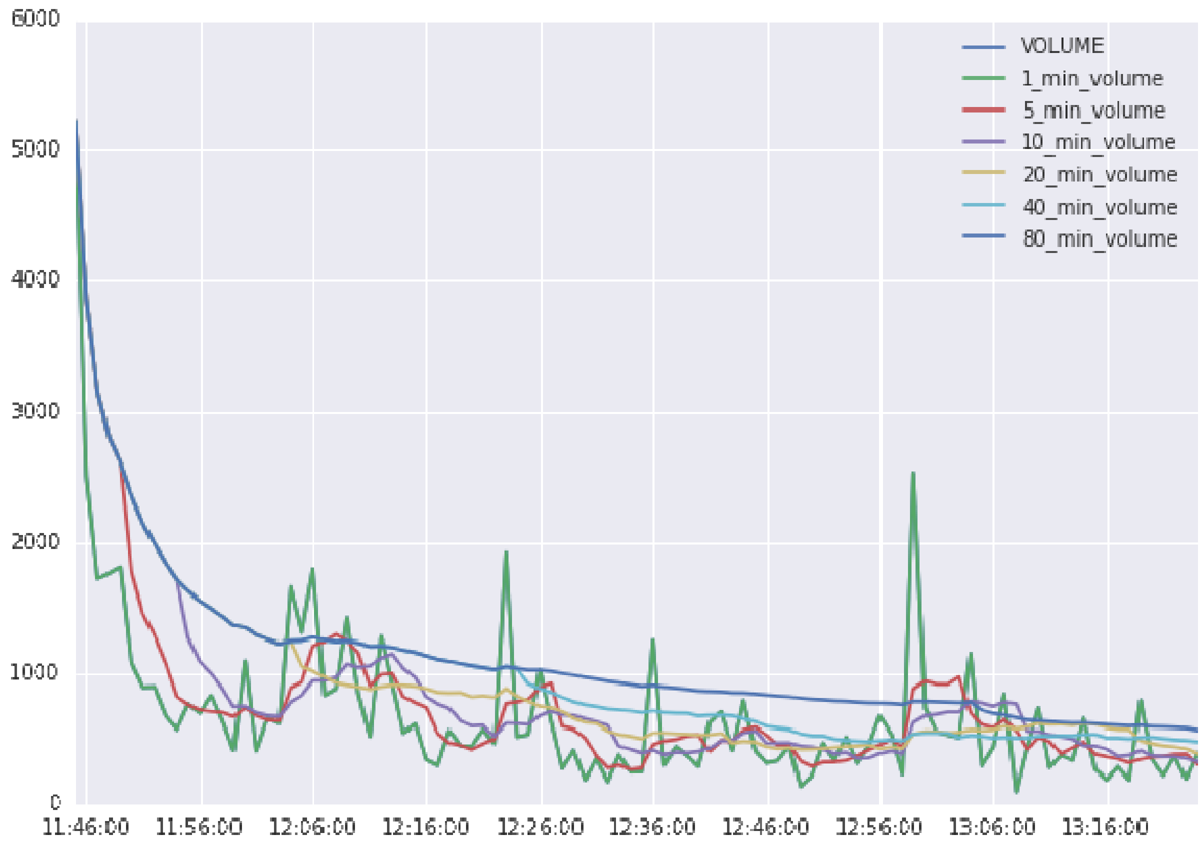


**Figure 11 – volume**

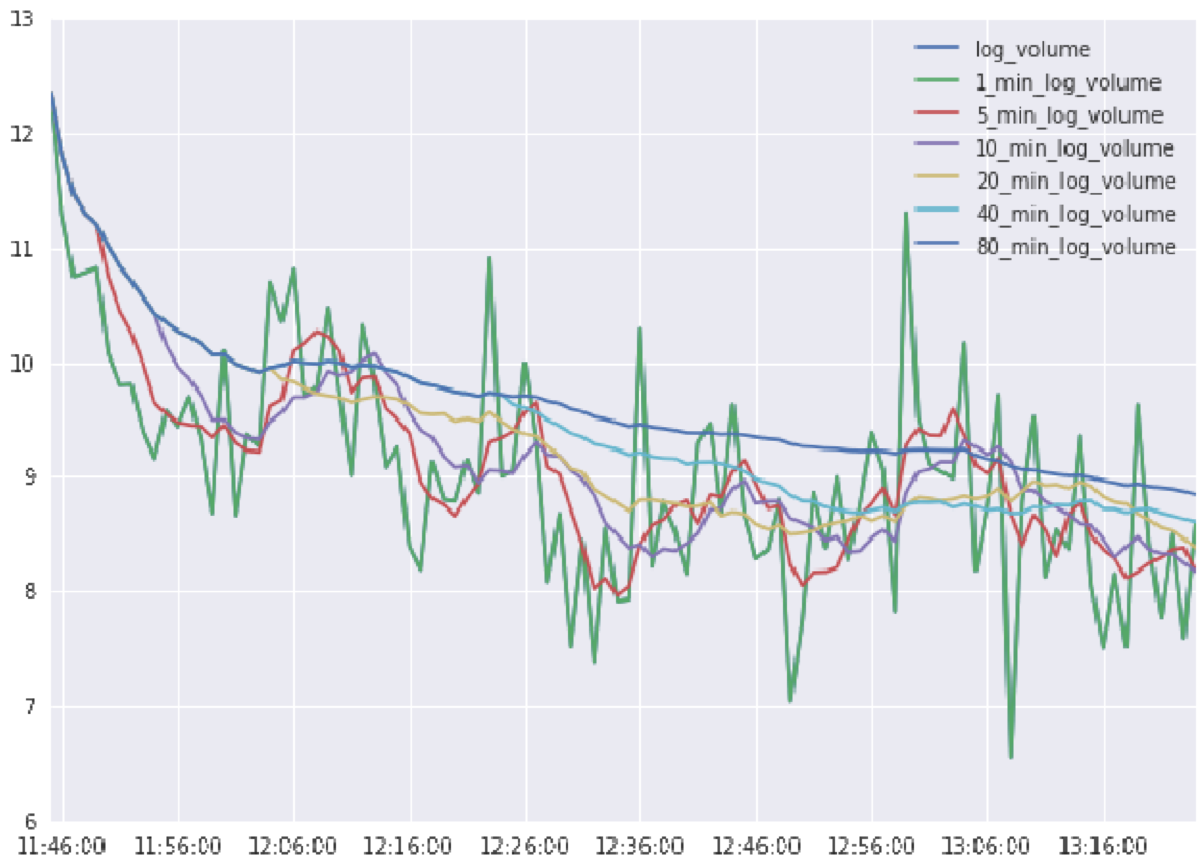


Figure 12 - log volume

Figure 11 and Figure 12 show examples of the volume and log volume columns. As volume varies quite wildly over each minute, log volume could be preferred as a feature as it has less extreme variations which may cause some analyses to be led astray.

We'll look at volume versus day of week and time of day. Figure 13 demonstrates that mean volume over x-minute periods does not fluctuate much by day of week. But, volume does vary considerably by hour, as seen in Figure 14. We can see in Figure 15 that even the log-volume has different distributions by hour. We'd probably want to produce a variable telling us how the current x-minute volume compares to some prior distribution of x-minute volumes, given the current hour of day. This is so that we can express in a meaningful day that 'x-minute volume was high, considering the current hour of day.' If we did not condition on hour of day, our indicator of 'high volume' would be true a lot near the beginning and end of day, and mostly false in the middle of the day. It might be useful to be able to indicate 'unusually high volume' as this, combined with some other indicator, could be predictive of the next move. E.g., if there is a big prior return on high volume, perhaps this move is likely to continue. But, if there is a big prior return on low volume, maybe this move is more likely to reverse. Figure 16 shows this 'time-adjusted' log-volume by hour of day. Clearly, the distribution is much more constant across hour of day; a high value on this variable does mean unusually high volume *given the current hour of day*. Note that we've not created this variable for the rows with time >= 1800, as this is not really relevant for our predictions, being volume at the closing minute / settlement minute.

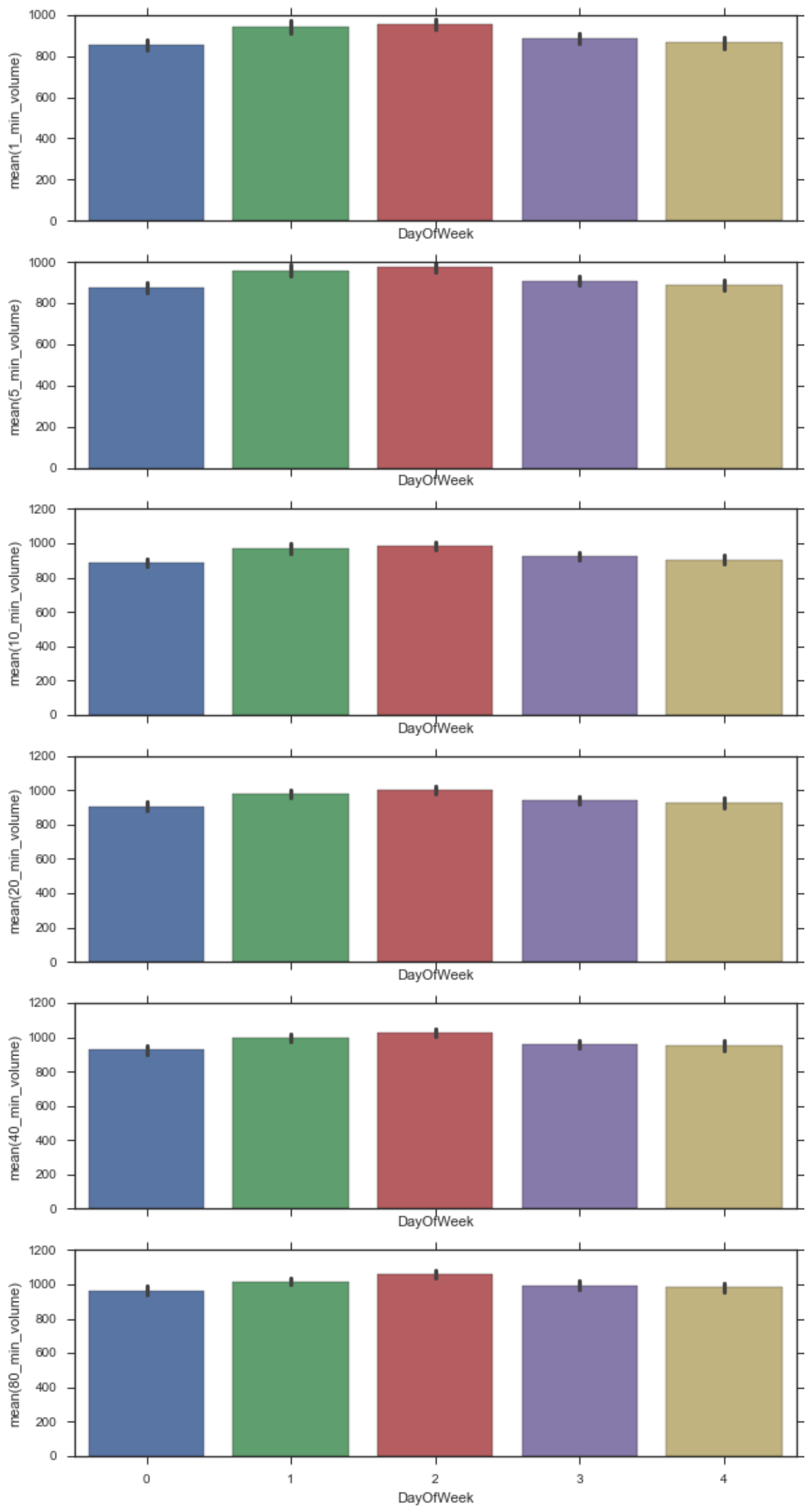


Figure 13 - Volume by day of week

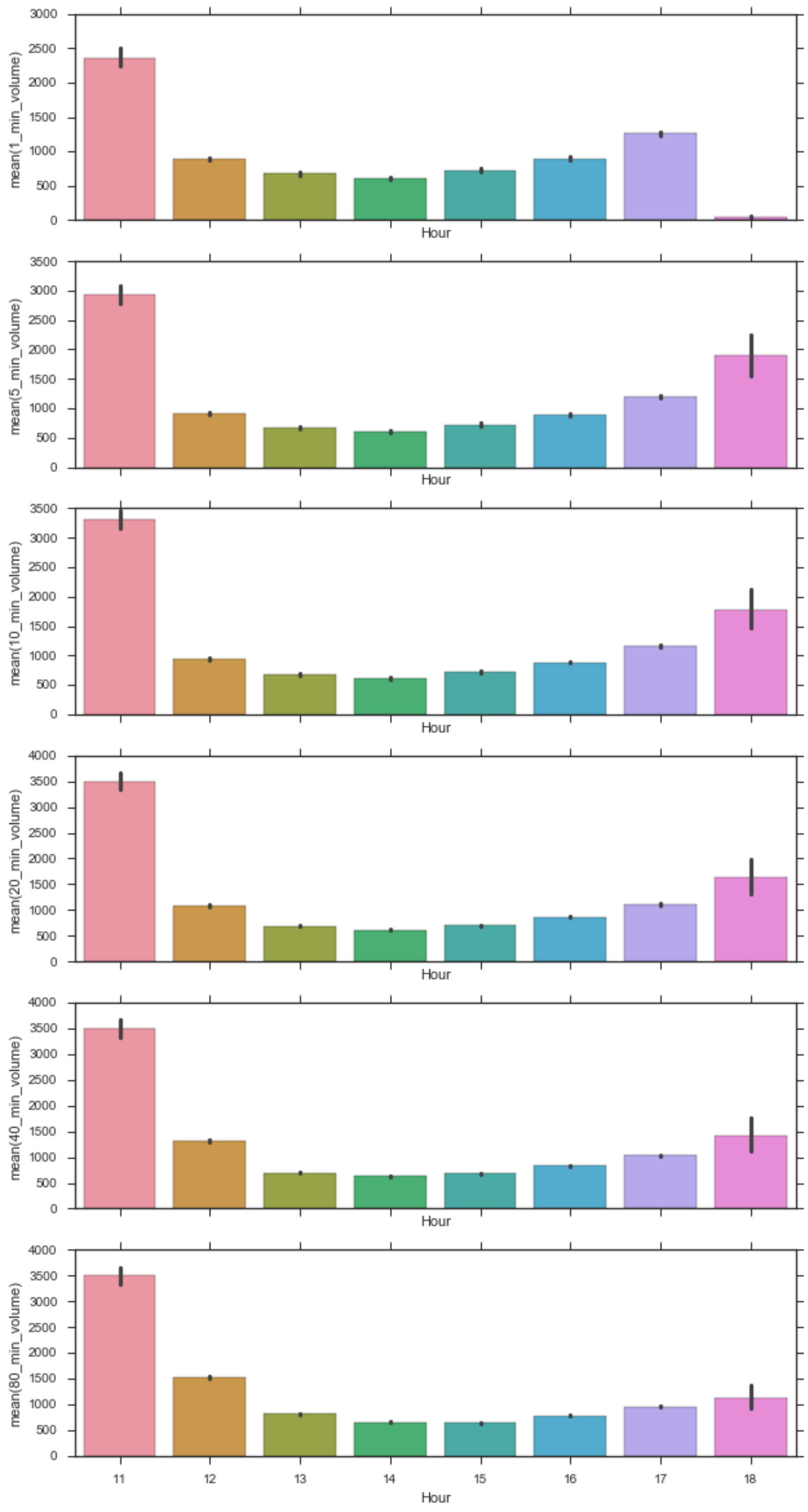


Figure 14 - Volume by hour

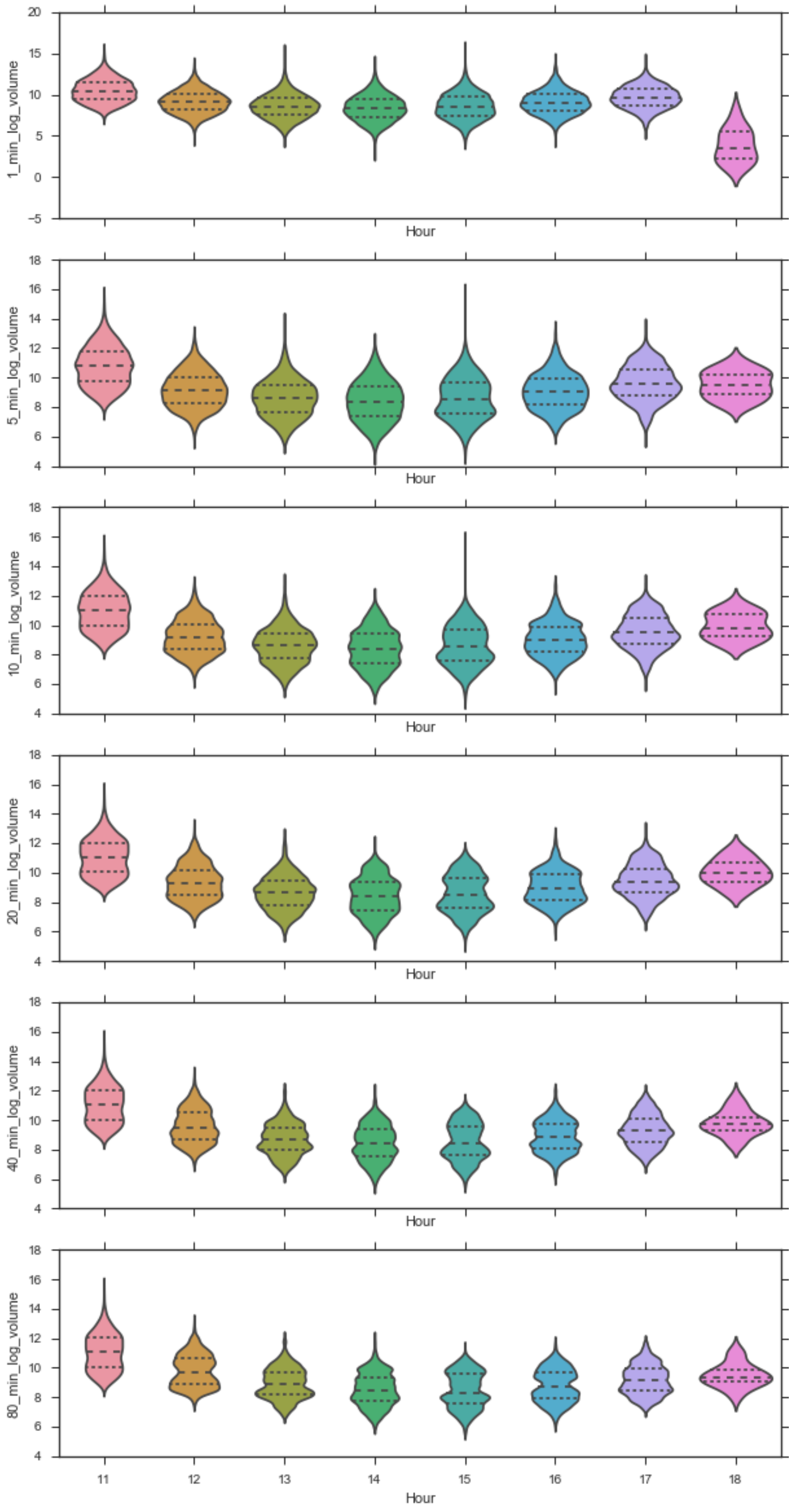


Figure 15 - log-volume distributions by hour

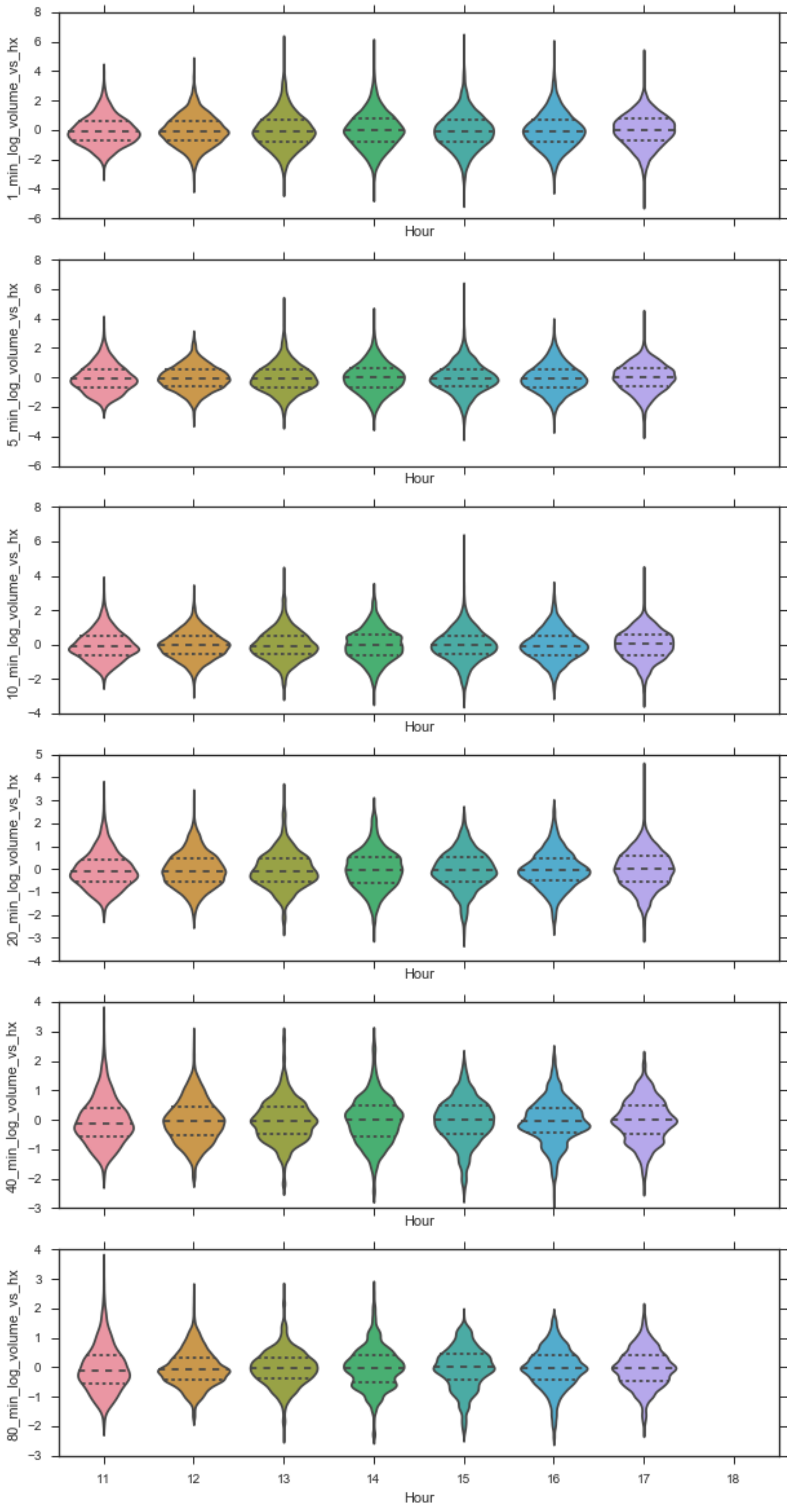


Figure 16 - time-adjusted log-volume by hour distributions

The next set of columns represent buying up / selling down behavior. This is, loosely, a measure of how much volume went behind a particular directional move. It's possible that upmoves on high volume are more predictive of subsequent upmoves than upmoves on low volume. So, we'll mark a minute as 'up' if its LAST_PRICE is higher than the LAST_PRICE in the previous minute. For the opening minute of the day, we'll compare LAST_PRICE to OPEN. Then, we sign the volume in that minute as the raw 'busd' value. E.g., if LAST_PRICE at 1148 is 8889 and LAST_PRICE at 1149 is 8888, and VOLUME in the 1149 minute is 1000, then 'busd' is -1000. i.e., we simply return the volume in an up minute, and -1 times the volume in a down minute. Then we average over x minutes. We'll do the same for log volume too. Again, log values may be better for some analyses, as seen in Figure 17 and Figure 18. Finally, we'll look at busd_time, which is our time-adjusted log-volume measure over 1-minute, multiplied by the sign of the 1-minute prior return, then averaged over 1-minute, 5-minutes, etc. An example for this is shown in Figure 19.

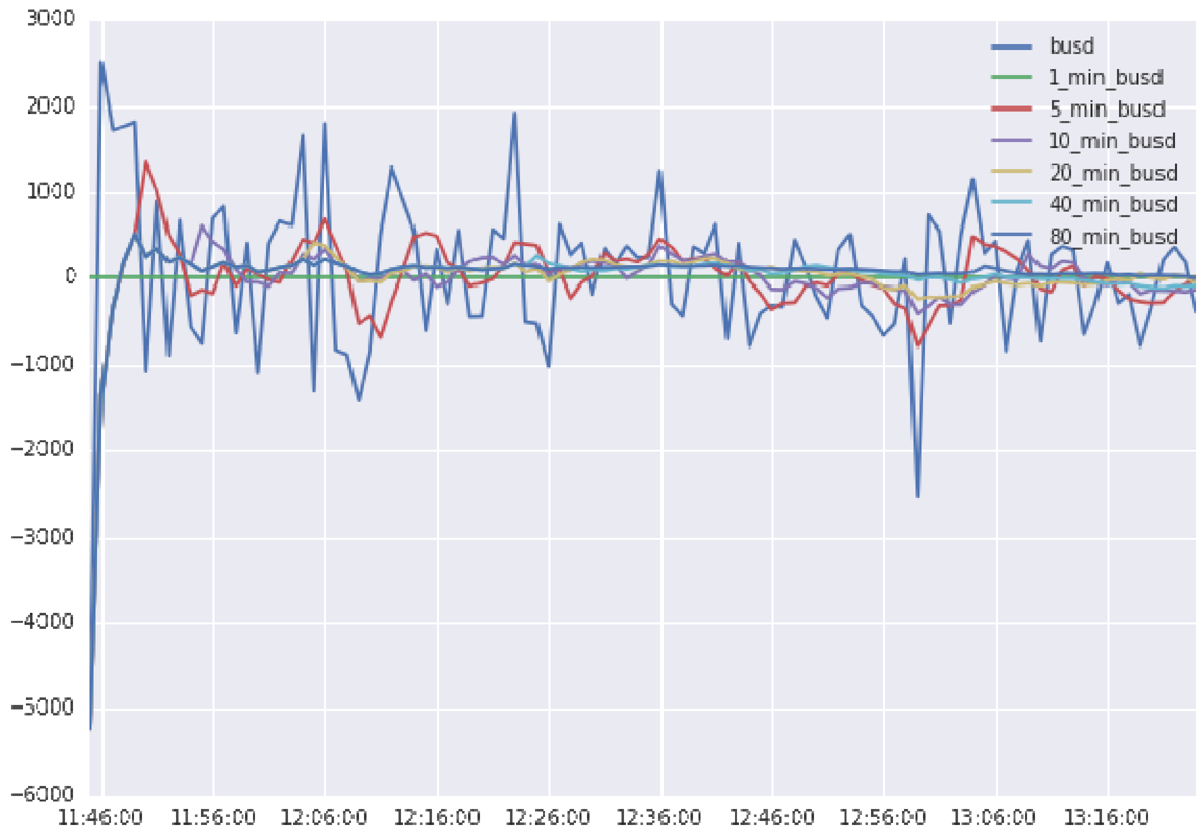


Figure 17 – buying up / selling down example

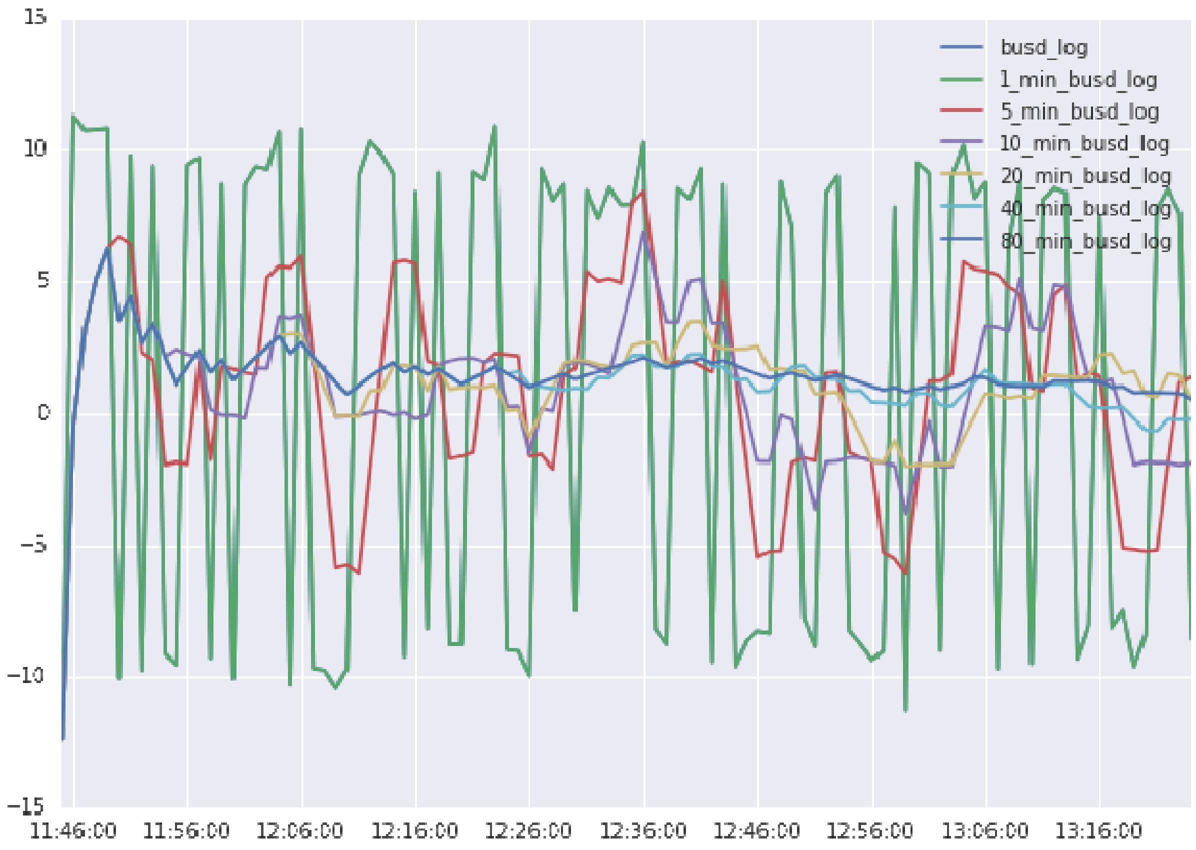


Figure 18 - log buying up / selling down example

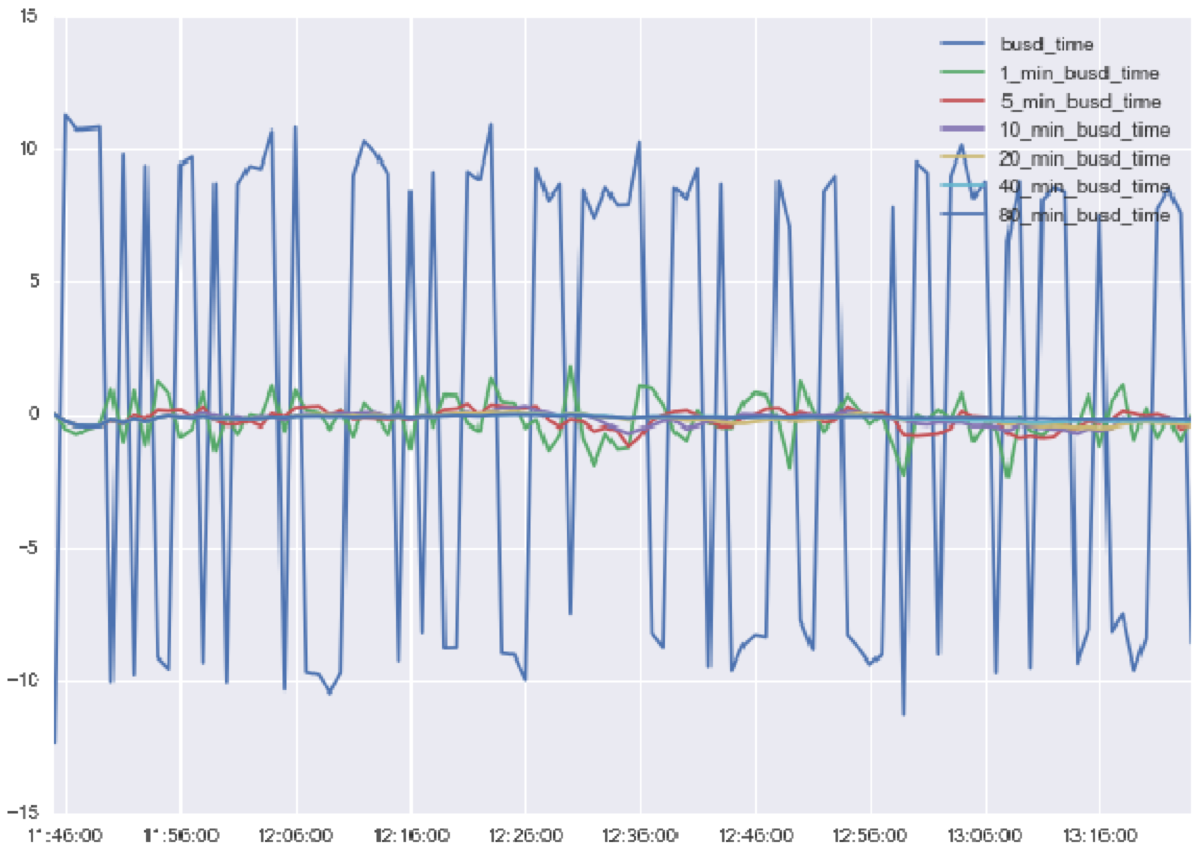


**Figure 19 - busd time example plot**

Next we look at high-low ranges over x-minute prior periods. To construct this we take the difference between the maximum of the x-minute HIGH less the minimum of the x-minute LOW. This may be useful as it proxies the recent level of price volatility. The distributions of these features are shown in Figure 20, they're heavy right tailed.

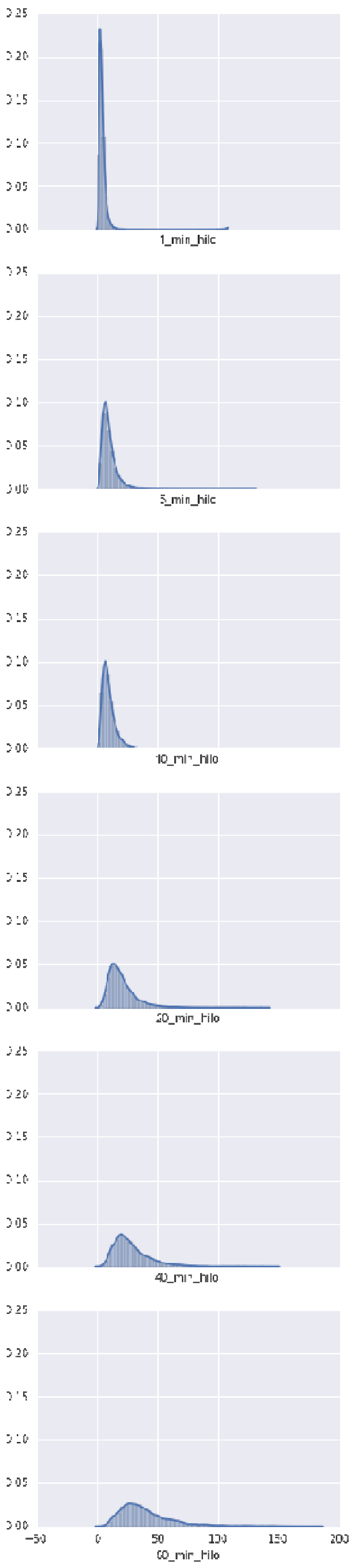


**Figure 20 - high-low range distributions**

The final set of indicators is the current LAST_PRICE relative to the 1-minute-ago x-minute high-low range. E.g., if at 1241 the LAST_PRICE is 8801, and between 1200 and 1240 the highest price was 8800 and the lowest price was 8700, then we return 1.01. We scale this such that if the current price matches the high, it will be 1; if it matches the low, it will be -1; if it's at the mean of the high and low, it will be 0; and so on. This can be a good proxy for breakouts of a range, which may suggest continuation of previous movements. Figure 21 provides an example at the 10-minute timeframe.

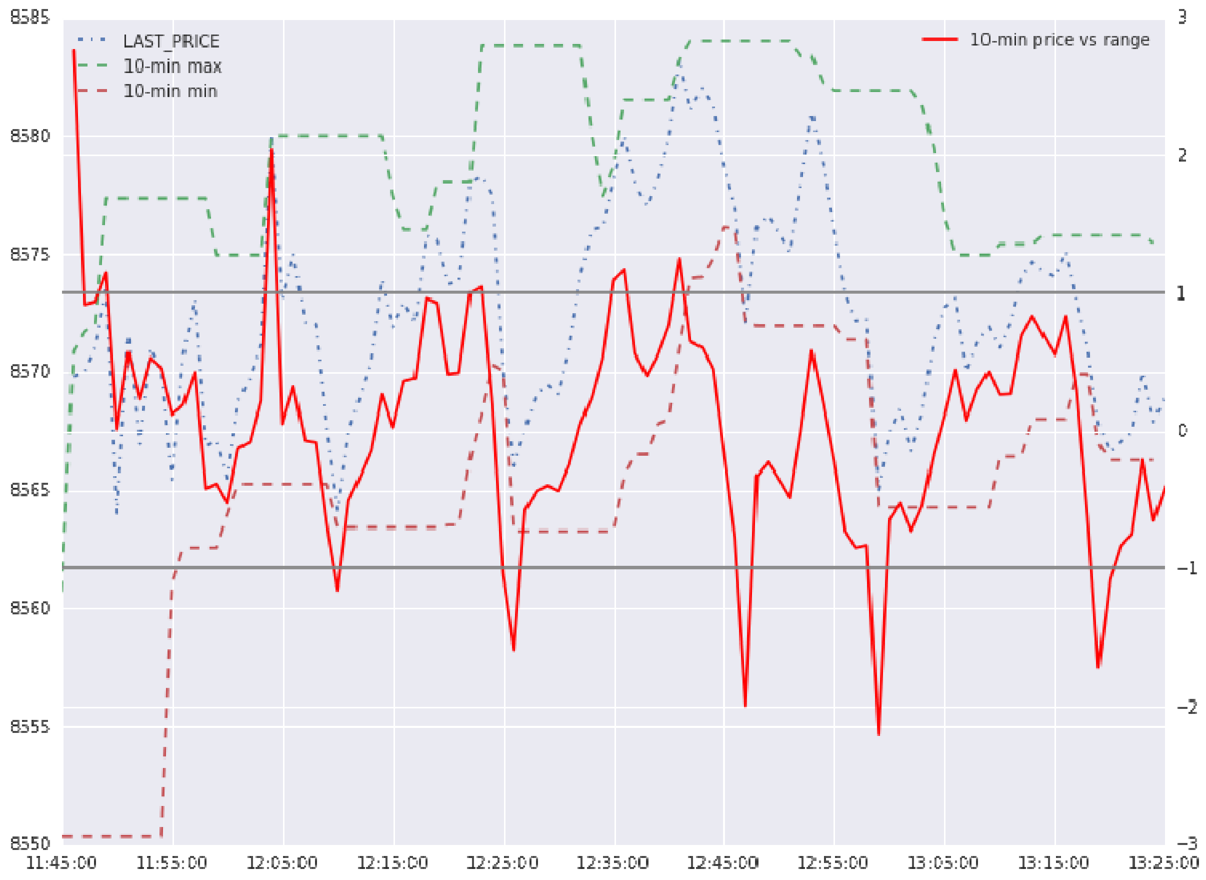


Figure 21 - Price vs range example

Exploratory Analysis

Figure 22 shows violinplots of the x-minute returns versus day of week. The width of the violin at a give return level indicates the density of the distribution i.e., returns near 0 are more common than extreme returns. There isn't any obvious pattern by day of week. There are some large returns i.e., the distribution is heavy tailed, but the magnitude of returns does not suggest outliers that should be removed. (e.g., 1.5% return within 1 minute is plausible, this is an extreme event but not unreasonable.)

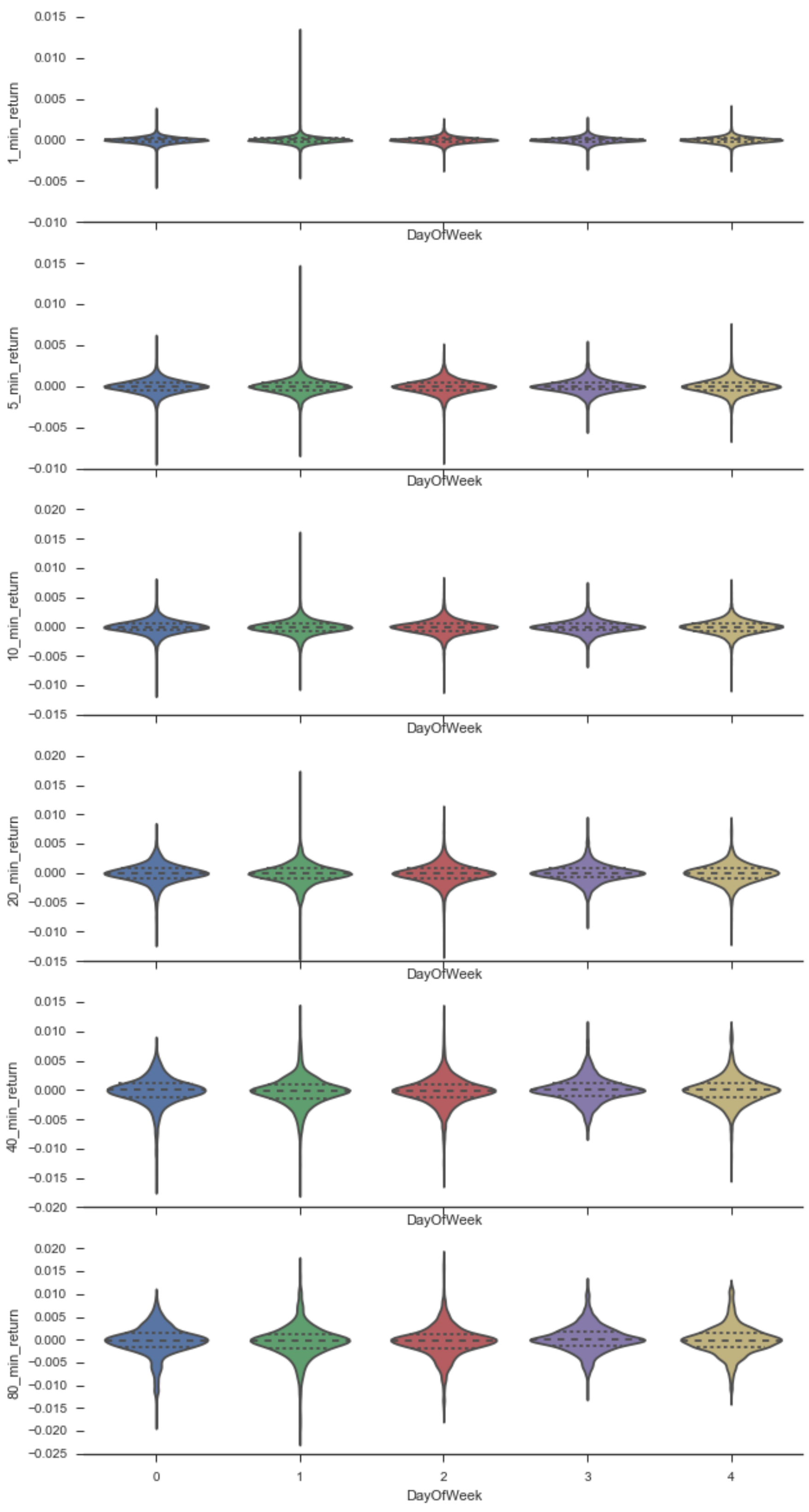


Figure 22 - return vs Day of Week

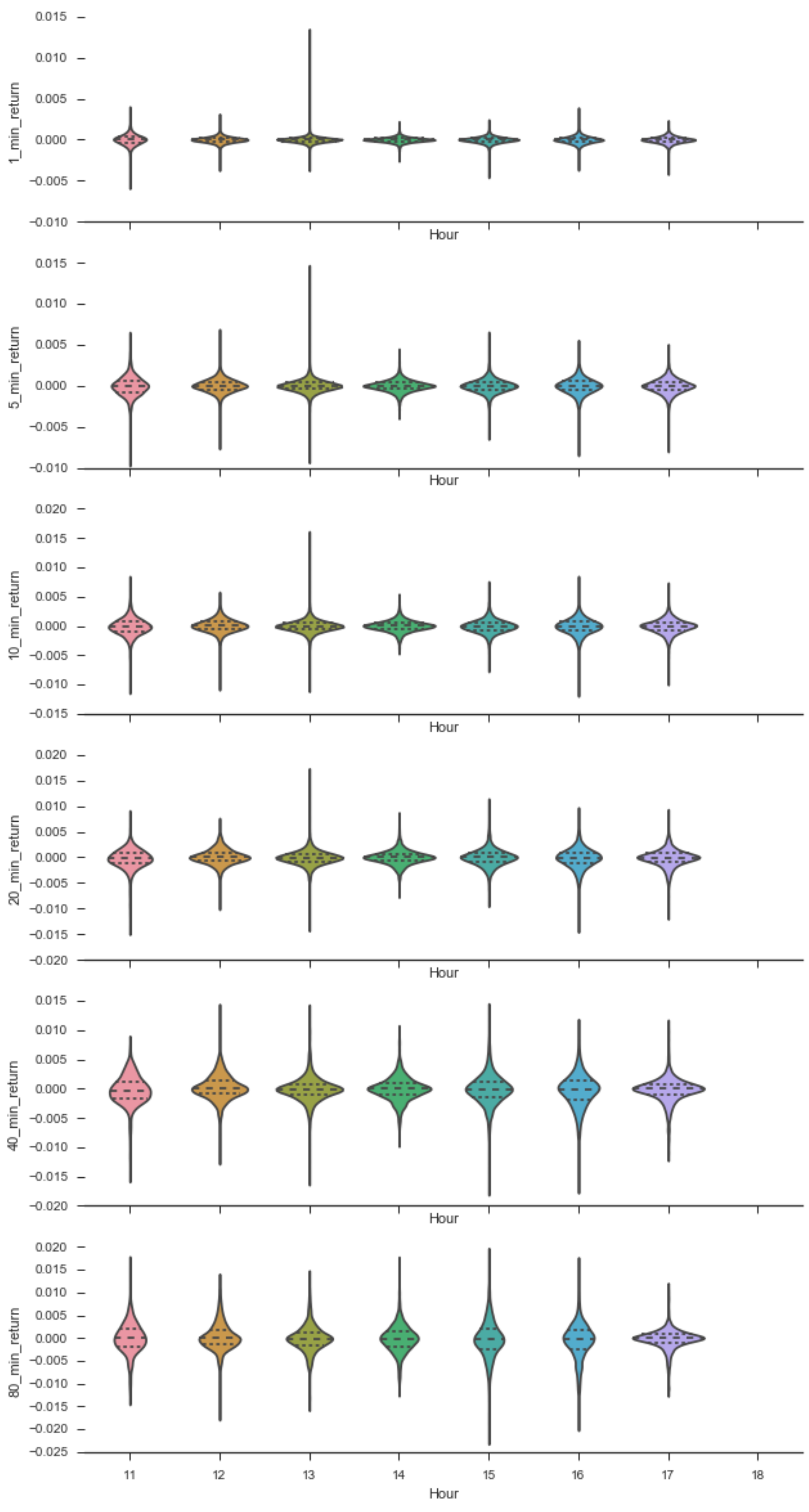


Figure 23 - x-minute return by hour

Figure 23 shows the distribution of x-minute returns by hour. There’s no trend of higher or lower returns by hour, but there may be some systematic tendency to have lower dispersion of returns nearer the middle of the day (around hour 14). i.e., returns are more variable near the open (hour 11) and close (hour 17).  Figure 24 explores this more closely – the heatmap shows the standard deviation of x-minute returns by hour of day. It’s reasonably clear that standard deviation of returns are lower in the middle of the day than at nearer the open or close.

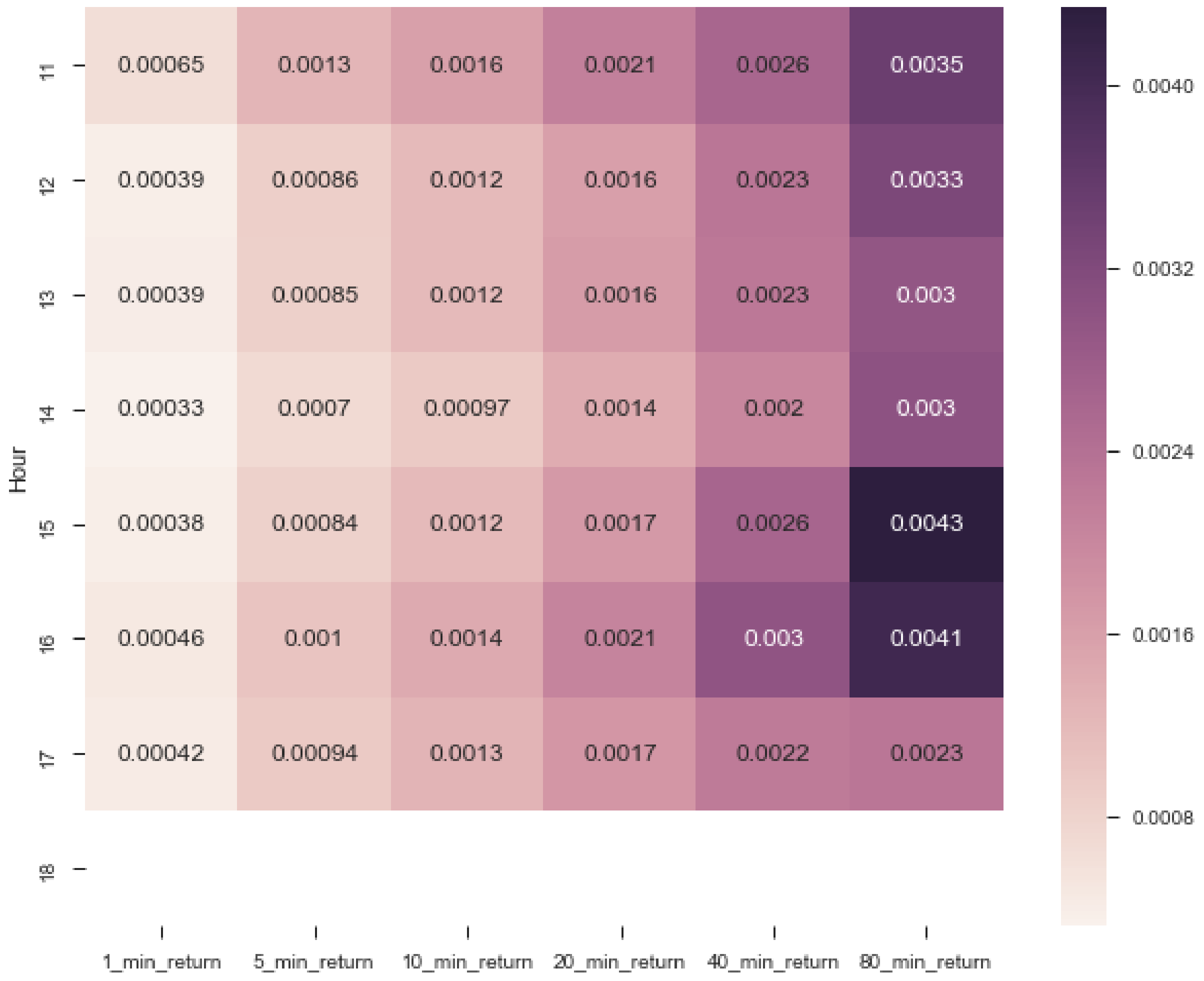


Figure 24 - standard deviation of x-minute returns by hour

Now we look at the relationship between x-minute prior return and x-minute (ahead) returns.

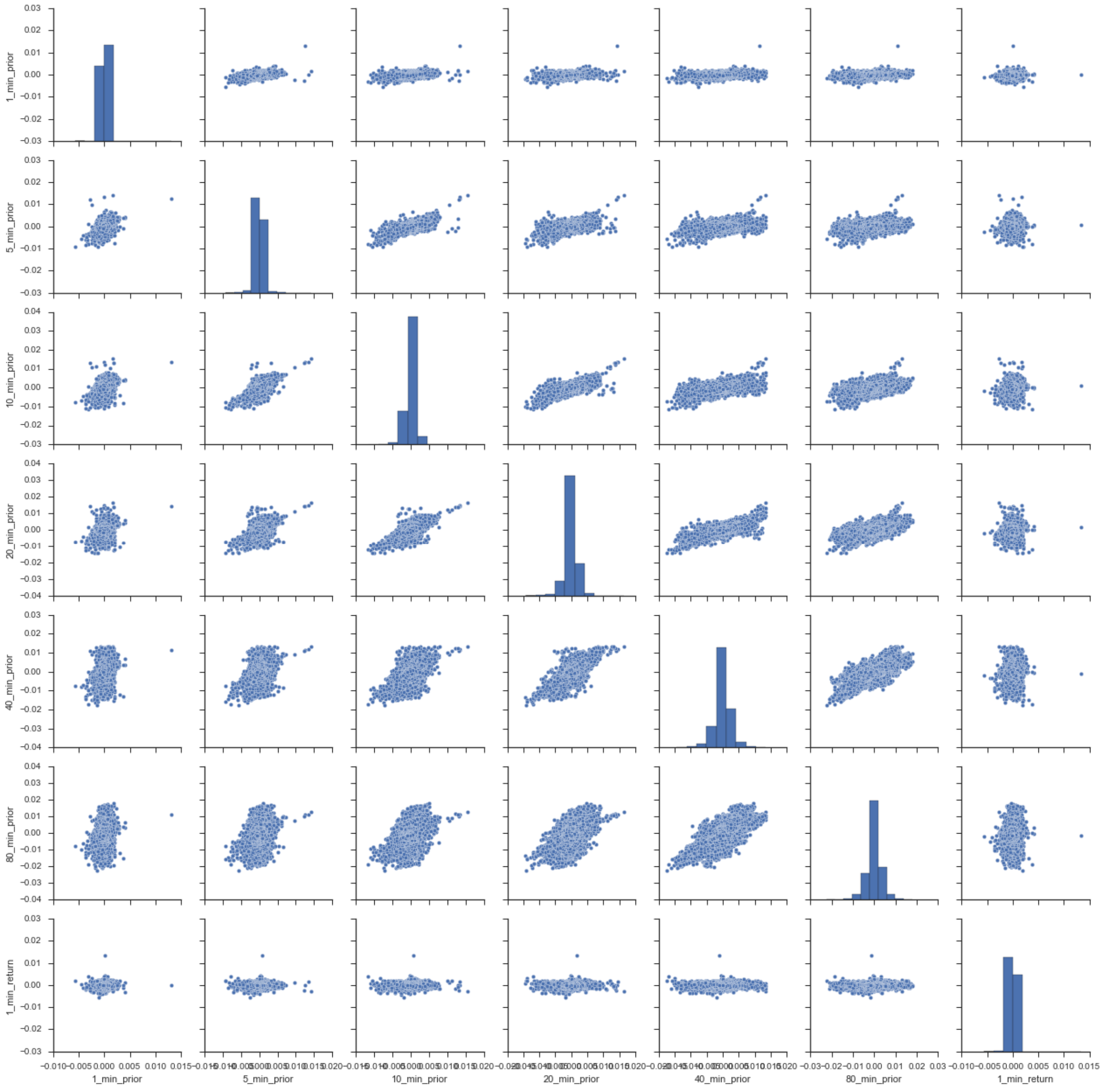


**Figure 25 - 1-minute return vs x-minute prior return scatterplot matrix**

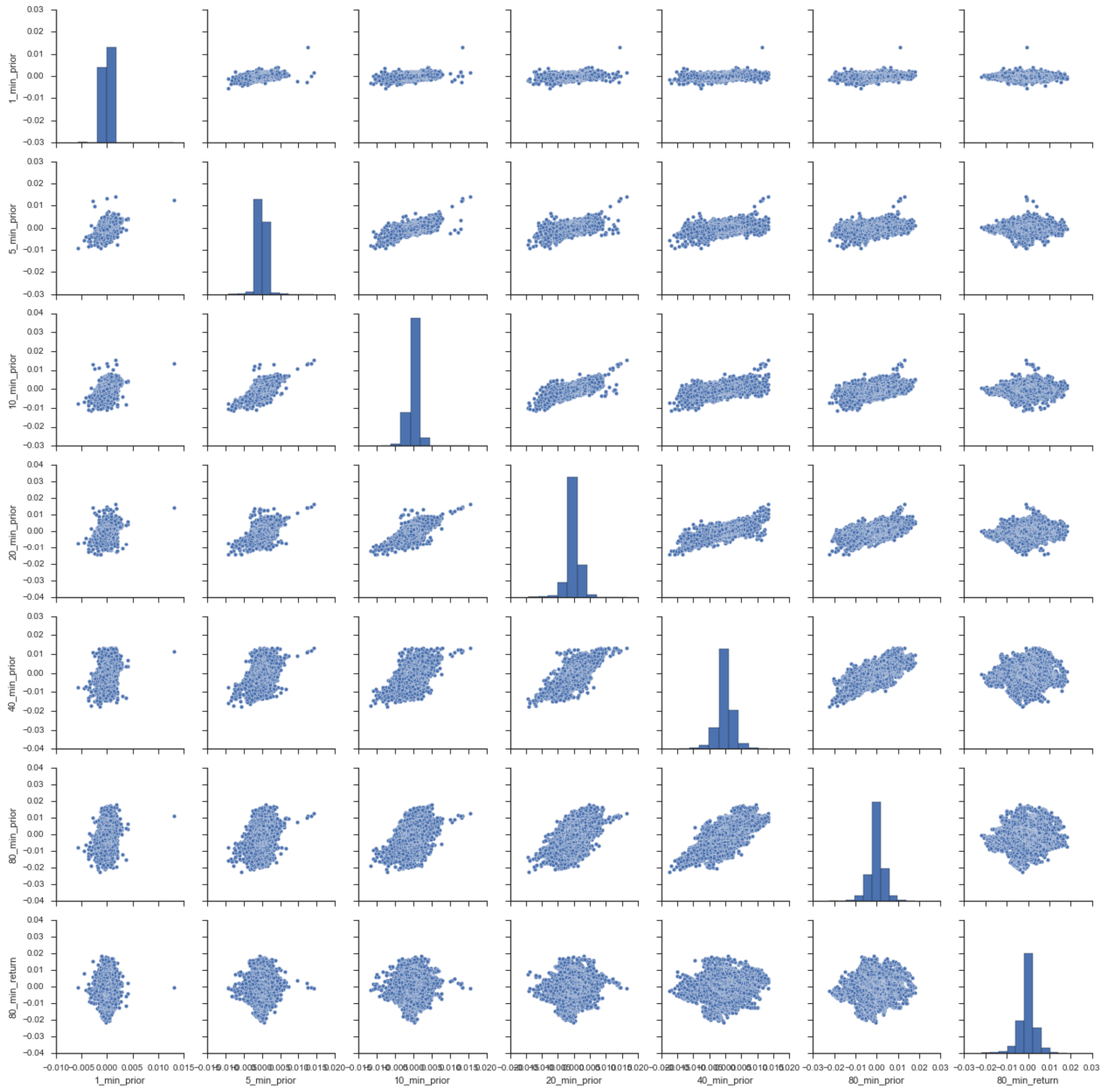


**Figure 26 - 80-minute return vs x-minute prior scatterplot matrix**

Figure 25 shows the 1-minute (ahead) return and all the x-minute prior returns in a scatterplot matrix. The x-minute prior returns are positively correlated with each other, which makes sense – if the 80-minute prior return was positive, it's much more likely that the 40-minute prior return was also positive, and so on. But, none of these x-minute prior returns seem to exhibit any visible pattern with the 1-minute return. Figure 26 shows the same scatterplot, but for 80-minute (ahead) returns instead. Here the scatterplots of x-minute prior returns versus 80-minute ahead returns are potentially more interesting, but the relationship is not clear. Figure 27 explores these relationships a little more closely – there is a small positive correlation between prior returns and 80-minute ahead returns; the longer the prior return the higher the correlation with the 80-minute ahead returns. This is consistent with a small momentum effect.

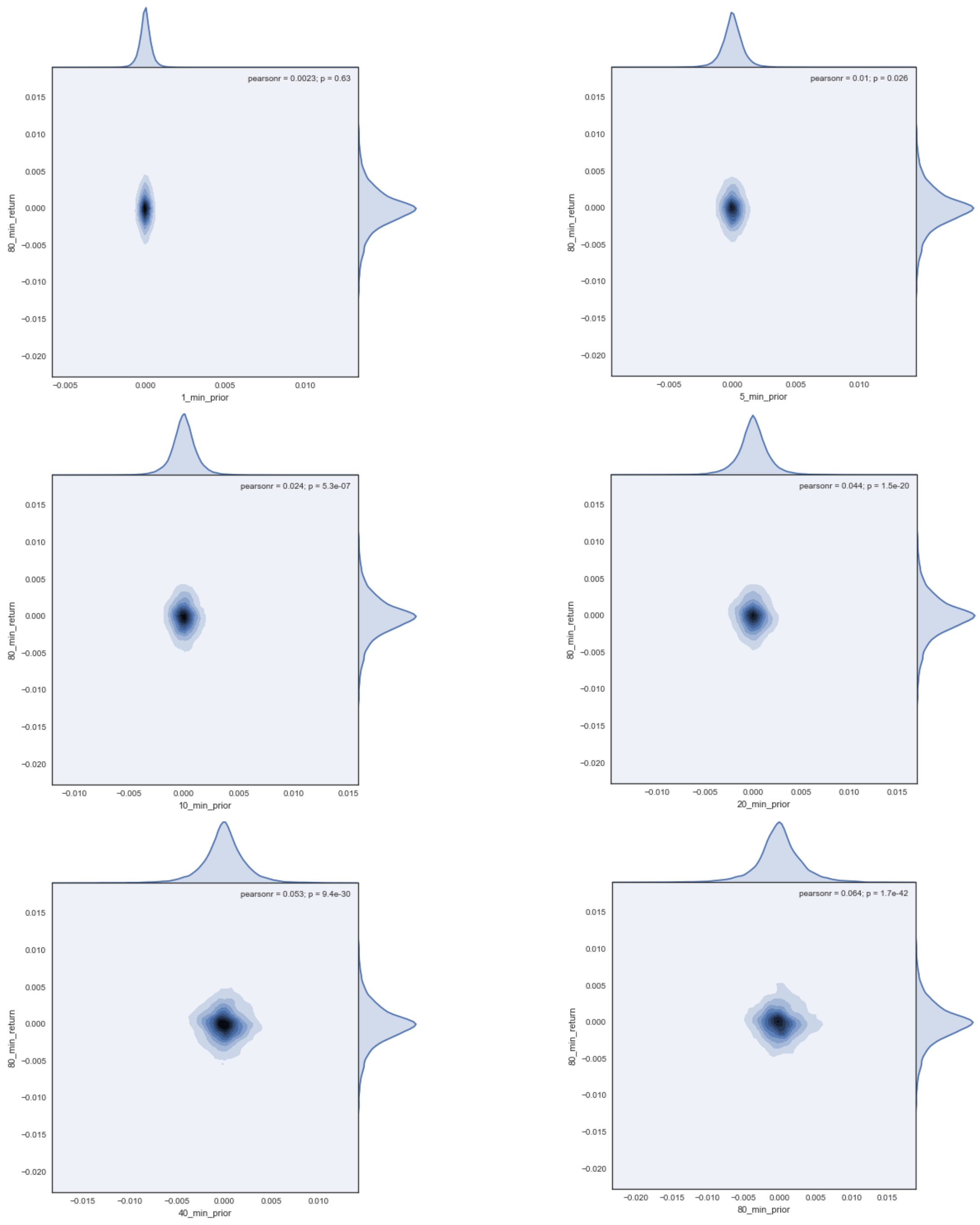


Figure 27 - kdeplots of x-minute prior vs 80-minute returns

Let’s explore this more generally, for all timescales. Figure 28 shows that there is a small negative correlation between short term prior returns and short term ahead returns, and a small positive relation between longer-term prior returns and longer-term ahead returns. i.e., in the short-term there is mean-reversion, and in the longer term there is momentum. These effects are quite small. But, the mean-reversion of short-term prior returns and momentum of longer-term returns is statistically significant, as seen in Figure 29.

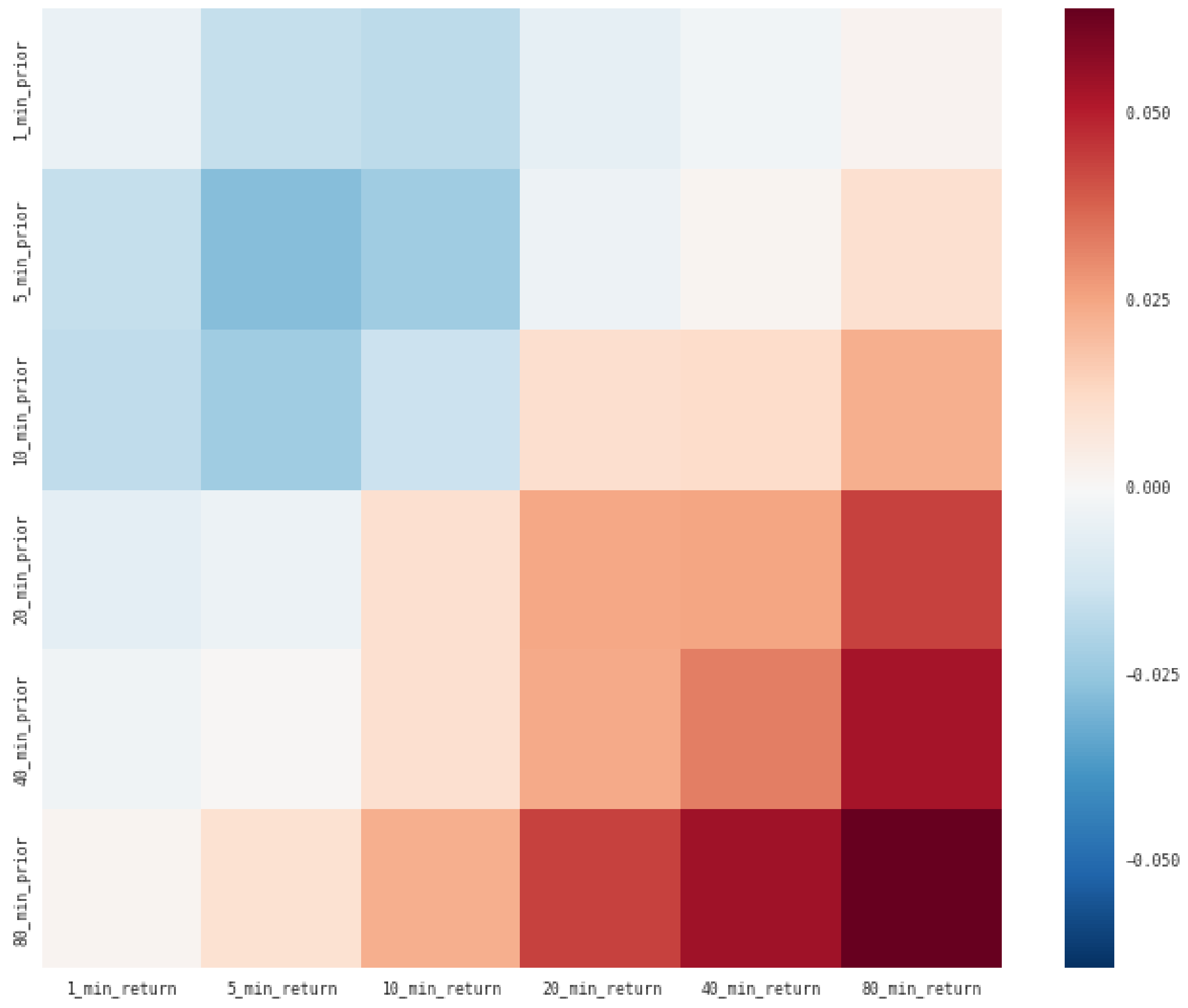


Figure 28 - heatmap of correlations between prior and ahead returns

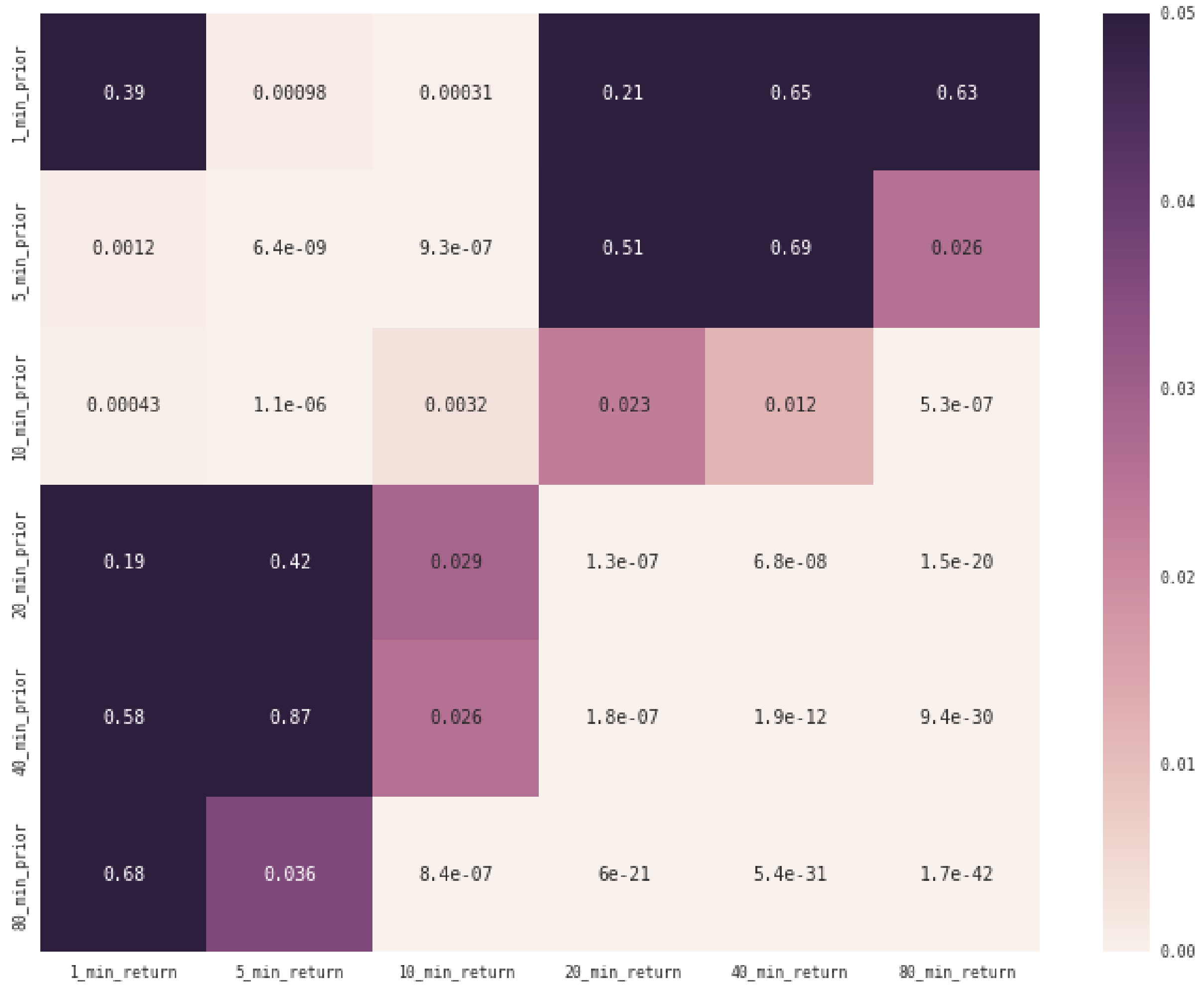
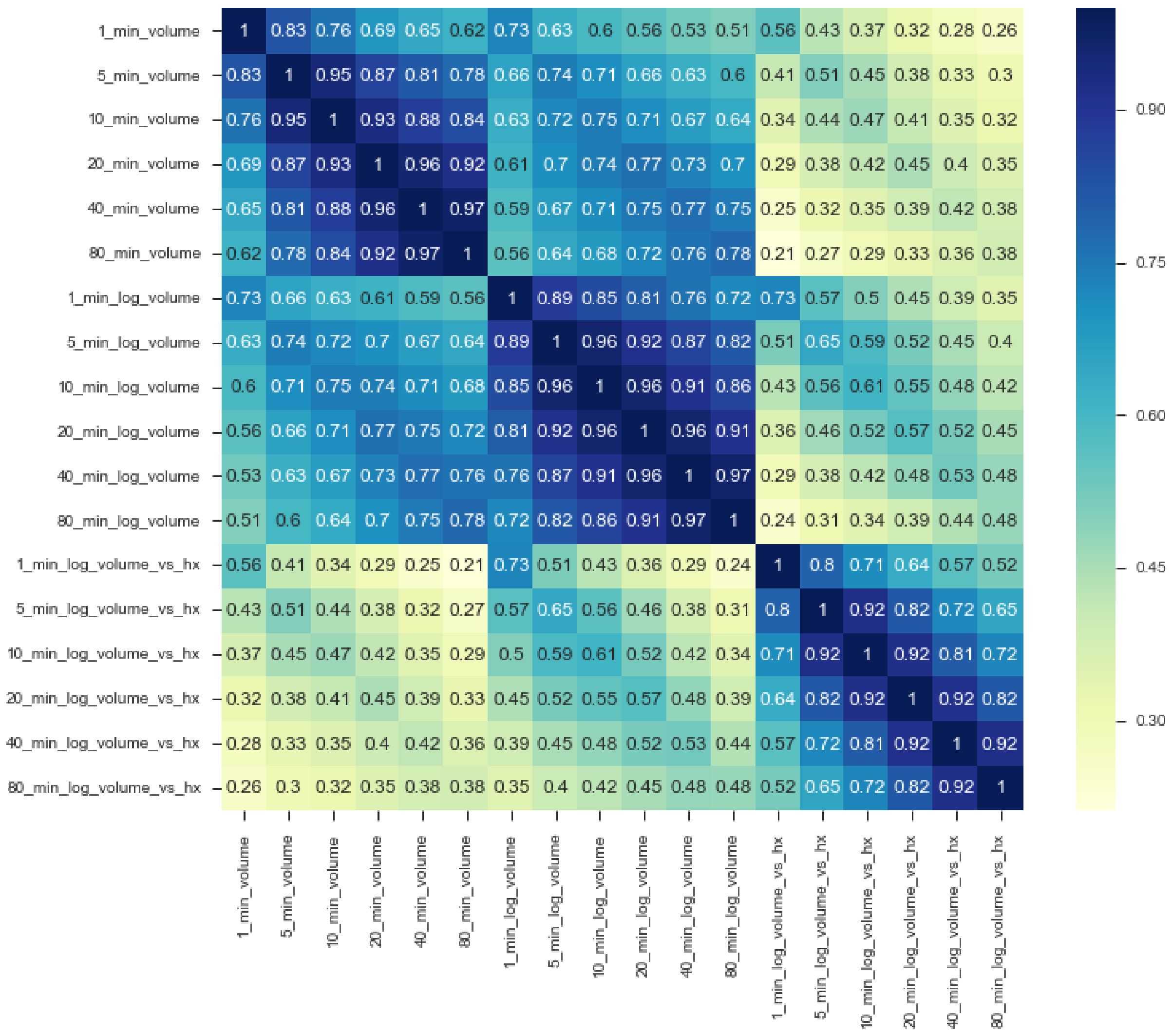


Figure 29 - p-value of correlations between x-minute prior and x-minute returns

Next we look at volume. Figure 30 shows a heatmap of the correlations between our various volume measures. All of these are positively correlated, which makes sense – they are all measuring whether volume is high or low. Within each measure, volumes at timescales that are closer together are more highly correlated (e.g., 40-minute and 80-minute volumes are more correlated than 1-minure and 80-minute volumes). The time-adjusted log-volume measures are least correlated to the raw volume measures.



| | 1_min_volume | 5_min_volume | 10_min_volume | 20_min_volume | 40_min_volume | 80_min_volume | 1_min_log_volume | 5_min_log_volume | 10_min_log_volume | 20_min_log_volume | 40_min_log_volume | 80_min_log_volume | 1_min_log_volume_vs_hx | 5_min_log_volume_vs_hx | 10_min_log_volume_vs_hx | 20_min_log_volume_vs_hx | 40_min_log_volume_vs_hx | 80_min_log_volume_vs_hx |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1_min_volume | 1 | 0.83 | 0.76 | 0.69 | 0.65 | 0.62 | 0.73 | 0.63 | 0.6 | 0.56 | 0.53 | 0.51 | 0.56 | 0.43 | 0.37 | 0.32 | 0.28 | 0.26 |
| 5_min_volume | 0.83 | 1 | 0.95 | 0.87 | 0.81 | 0.78 | 0.66 | 0.74 | 0.71 | 0.66 | 0.63 | 0.6 | 0.41 | 0.51 | 0.45 | 0.38 | 0.33 | 0.3 |
| 10_min_volume | 0.76 | 0.95 | 1 | 0.93 | 0.88 | 0.84 | 0.63 | 0.72 | 0.75 | 0.71 | 0.67 | 0.64 | 0.34 | 0.44 | 0.47 | 0.41 | 0.35 | 0.32 |
| 20_min_volume | 0.69 | 0.87 | 0.93 | 1 | 0.96 | 0.92 | 0.61 | 0.7 | 0.74 | 0.77 | 0.73 | 0.7 | 0.29 | 0.38 | 0.42 | 0.45 | 0.4 | 0.35 |
| 40_min_volume | 0.65 | 0.81 | 0.88 | 0.96 | 1 | 0.97 | 0.59 | 0.67 | 0.71 | 0.75 | 0.77 | 0.75 | 0.25 | 0.32 | 0.35 | 0.39 | 0.42 | 0.38 |
| 80_min_volume | 0.62 | 0.78 | 0.84 | 0.92 | 0.97 | 1 | 0.56 | 0.64 | 0.68 | 0.72 | 0.76 | 0.78 | 0.21 | 0.27 | 0.29 | 0.33 | 0.36 | 0.38 |
| 1_min_log_volume | 0.73 | 0.66 | 0.63 | 0.61 | 0.59 | 0.56 | 1 | 0.89 | 0.85 | 0.81 | 0.76 | 0.72 | 0.73 | 0.57 | 0.5 | 0.45 | 0.39 | 0.35 |
| 5_min_log_volume | 0.63 | 0.74 | 0.72 | 0.7 | 0.67 | 0.64 | 0.89 | 1 | 0.96 | 0.92 | 0.87 | 0.82 | 0.51 | 0.65 | 0.59 | 0.52 | 0.45 | 0.4 |
| 10_min_log_volume | 0.6 | 0.71 | 0.75 | 0.74 | 0.71 | 0.68 | 0.85 | 0.96 | 1 | 0.96 | 0.91 | 0.86 | 0.43 | 0.56 | 0.61 | 0.55 | 0.48 | 0.42 |
| 20_min_log_volume | 0.56 | 0.66 | 0.71 | 0.77 | 0.75 | 0.72 | 0.81 | 0.92 | 0.96 | 1 | 0.96 | 0.91 | 0.36 | 0.46 | 0.52 | 0.57 | 0.52 | 0.45 |
| 40_min_log_volume | 0.53 | 0.63 | 0.67 | 0.73 | 0.77 | 0.76 | 0.76 | 0.87 | 0.91 | 0.96 | 1 | 0.97 | 0.29 | 0.38 | 0.42 | 0.48 | 0.53 | 0.48 |
| 80_min_log_volume | 0.51 | 0.6 | 0.64 | 0.7 | 0.75 | 0.78 | 0.72 | 0.82 | 0.86 | 0.91 | 0.97 | 1 | 0.24 | 0.31 | 0.34 | 0.39 | 0.44 | 0.48 |
| 1_min_log_volume_vs_hx | 0.56 | 0.41 | 0.34 | 0.29 | 0.25 | 0.21 | 0.73 | 0.51 | 0.43 | 0.36 | 0.29 | 0.24 | 1 | 0.8 | 0.71 | 0.64 | 0.57 | 0.52 |
| 5_min_log_volume_vs_hx | 0.43 | 0.51 | 0.44 | 0.38 | 0.32 | 0.27 | 0.57 | 0.65 | 0.56 | 0.46 | 0.38 | 0.31 | 0.8 | 1 | 0.92 | 0.82 | 0.72 | 0.65 |
| 10_min_log_volume_vs_hx | 0.37 | 0.45 | 0.47 | 0.42 | 0.35 | 0.29 | 0.5 | 0.59 | 0.61 | 0.52 | 0.42 | 0.34 | 0.71 | 0.92 | 1 | 0.92 | 0.81 | 0.72 |
| 20_min_log_volume_vs_hx | 0.32 | 0.38 | 0.41 | 0.45 | 0.39 | 0.33 | 0.45 | 0.52 | 0.55 | 0.57 | 0.48 | 0.39 | 0.64 | 0.82 | 0.92 | 1 | 0.92 | 0.82 |
| 40_min_log_volume_vs_hx | 0.28 | 0.33 | 0.35 | 0.4 | 0.42 | 0.36 | 0.39 | 0.45 | 0.48 | 0.52 | 0.53 | 0.44 | 0.57 | 0.72 | 0.81 | 0.92 | 1 | 0.92 |
| 80_min_log_volume_vs_hx | 0.26 | 0.3 | 0.32 | 0.35 | 0.38 | 0.38 | 0.35 | 0.4 | 0.42 | 0.45 | 0.48 | 0.48 | 0.52 | 0.65 | 0.72 | 0.82 | 0.92 | 1 |

Figure 30 - Correlation heatmap for various volume measures

There isn't any reason to expect that volume should directly correlate with returns. Figure 31 demonstrates this quite clearly.

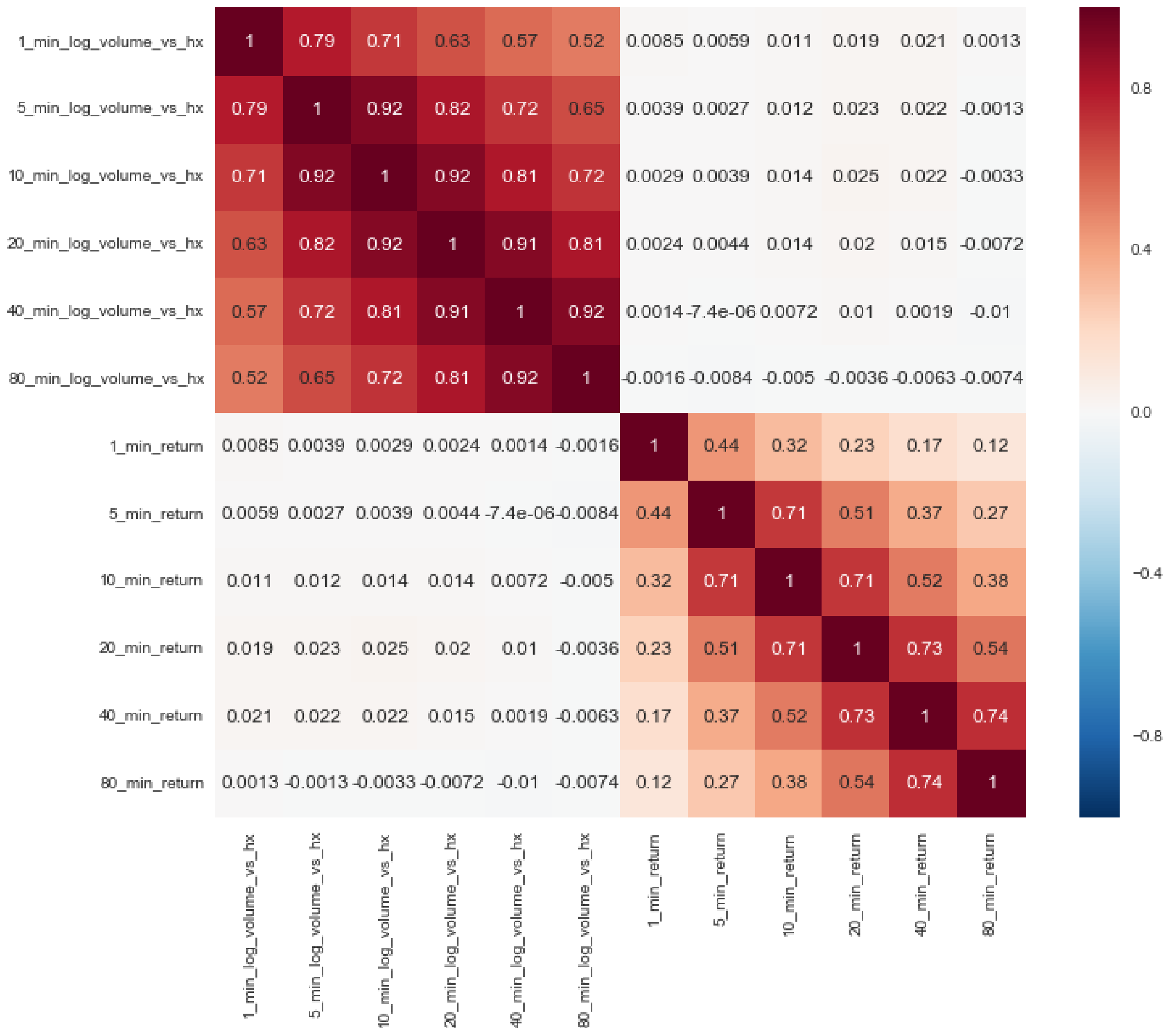


Figure 31 - time-adjusted log-volumes vs x-minute returns

We'll look at the interaction between volume, prior moves, and returns. Figure 32 suggests high time-adjusted 1-minute volume and negative prior 1-minute leads to positive 1-minute returns; Figure 33 suggests high time-adjusted 80-minute volume and positive prior 80-minute leads to positive 80-minute returns. Interestingly, Figure 33 may also be suggesting that a big 80-minute prior move on low time-adjusted 80-minute log-volume may lead to a reversal.

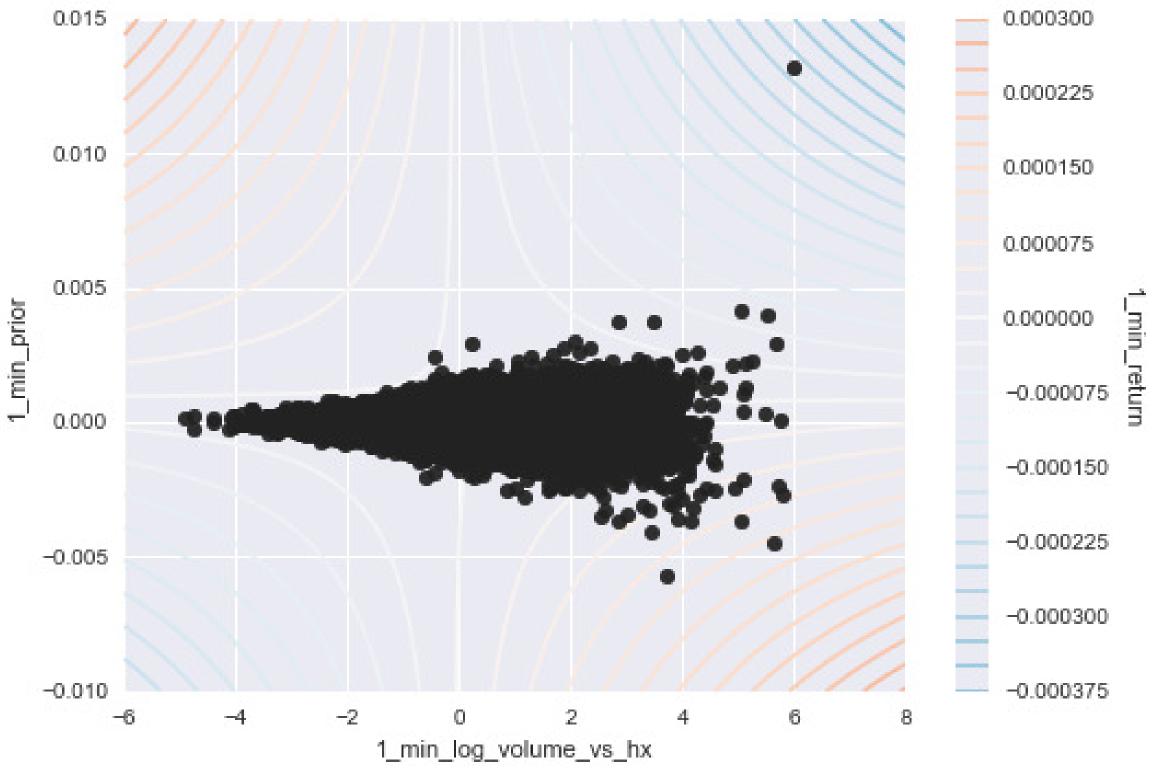


Figure 32 - volume and prior interaction on 1 minute returns

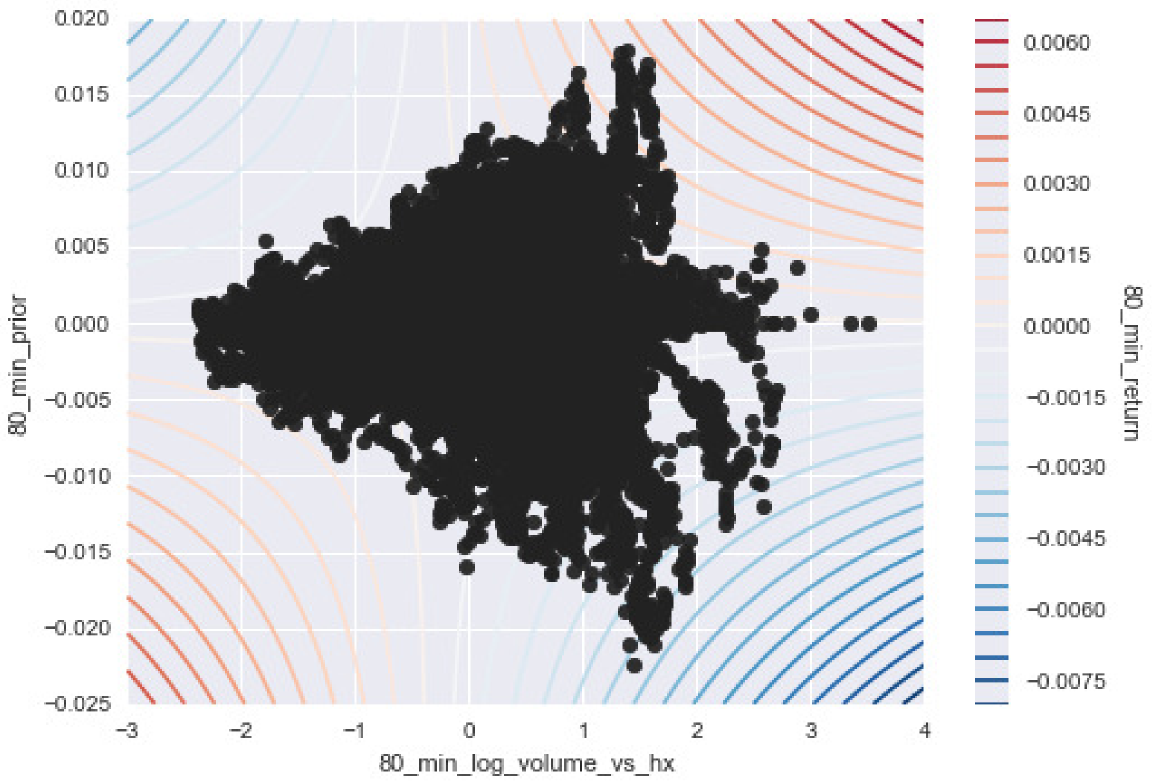
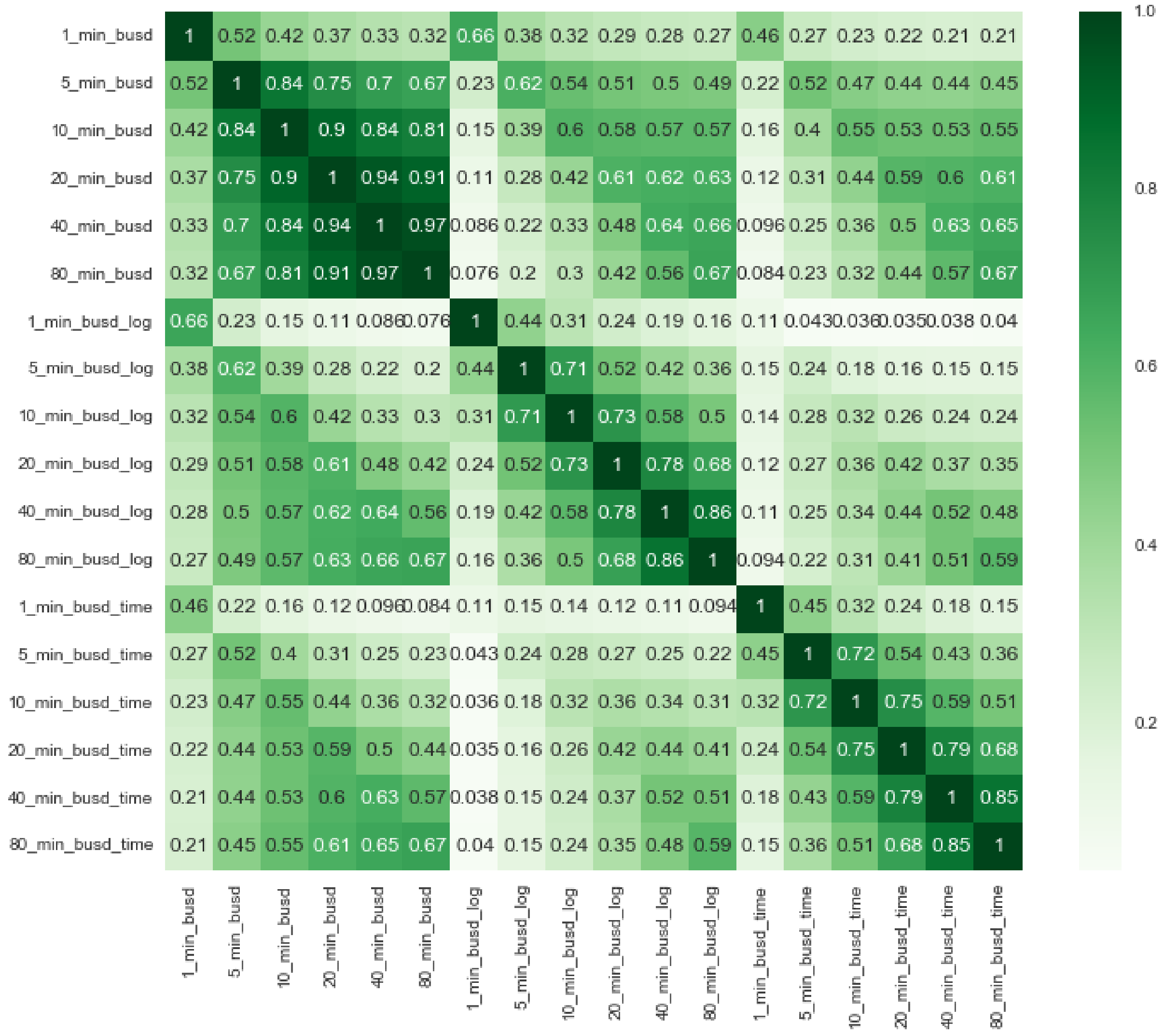


Figure 33 - volume and prior interaction on 80 minute returns

Now we look at the buying up / selling down indicators. Figure 34 shows that these are all positively correlated with each other, which is reassuring as they are all measuring a similar idea. Longer term indicators are more correlated with each other than shorter term indicators. The various styles of indicators can be relatively different from each other (though they are positively correlated).



| | 1_min_busd | 5_min_busd | 10_min_busd | 20_min_busd | 40_min_busd | 80_min_busd | 1_min_busd_log | 5_min_busd_log | 10_min_busd_log | 20_min_busd_log | 40_min_busd_log | 80_min_busd_log | 1_min_busd_time | 5_min_busd_time | 10_min_busd_time | 20_min_busd_time | 40_min_busd_time | 80_min_busd_time |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1_min_busd | 1 | 0.52 | 0.42 | 0.37 | 0.33 | 0.32 | 0.66 | 0.38 | 0.32 | 0.29 | 0.28 | 0.27 | 0.46 | 0.27 | 0.23 | 0.22 | 0.21 | 0.21 |
| 5_min_busd | 0.52 | 1 | 0.84 | 0.75 | 0.7 | 0.67 | 0.23 | 0.62 | 0.54 | 0.51 | 0.5 | 0.49 | 0.22 | 0.52 | 0.47 | 0.44 | 0.44 | 0.45 |
| 10_min_busd | 0.42 | 0.84 | 1 | 0.9 | 0.84 | 0.81 | 0.15 | 0.39 | 0.6 | 0.58 | 0.57 | 0.57 | 0.16 | 0.4 | 0.55 | 0.53 | 0.53 | 0.55 |
| 20_min_busd | 0.37 | 0.75 | 0.9 | 1 | 0.94 | 0.91 | 0.11 | 0.28 | 0.42 | 0.61 | 0.62 | 0.63 | 0.12 | 0.31 | 0.44 | 0.59 | 0.6 | 0.61 |
| 40_min_busd | 0.33 | 0.7 | 0.84 | 0.94 | 1 | 0.97 | 0.086 | 0.22 | 0.33 | 0.48 | 0.64 | 0.66 | 0.096 | 0.25 | 0.36 | 0.5 | 0.63 | 0.65 |
| 80_min_busd | 0.32 | 0.67 | 0.81 | 0.91 | 0.97 | 1 | 0.076 | 0.2 | 0.3 | 0.42 | 0.56 | 0.67 | 0.084 | 0.23 | 0.32 | 0.44 | 0.57 | 0.67 |
| 1_min_busd_log | 0.66 | 0.23 | 0.15 | 0.11 | 0.086 | 0.076 | 1 | 0.44 | 0.31 | 0.24 | 0.19 | 0.16 | 0.11 | 0.043 | 0.036 | 0.035 | 0.038 | 0.04 |
| 5_min_busd_log | 0.38 | 0.62 | 0.39 | 0.28 | 0.22 | 0.2 | 0.44 | 1 | 0.71 | 0.52 | 0.42 | 0.36 | 0.15 | 0.24 | 0.18 | 0.16 | 0.15 | 0.15 |
| 10_min_busd_log | 0.32 | 0.54 | 0.6 | 0.42 | 0.33 | 0.3 | 0.31 | 0.71 | 1 | 0.73 | 0.58 | 0.5 | 0.14 | 0.28 | 0.32 | 0.26 | 0.24 | 0.24 |
| 20_min_busd_log | 0.29 | 0.51 | 0.58 | 0.61 | 0.48 | 0.42 | 0.24 | 0.52 | 0.73 | 1 | 0.78 | 0.68 | 0.12 | 0.27 | 0.36 | 0.42 | 0.37 | 0.35 |
| 40_min_busd_log | 0.28 | 0.5 | 0.57 | 0.62 | 0.64 | 0.56 | 0.19 | 0.42 | 0.58 | 0.78 | 1 | 0.86 | 0.11 | 0.25 | 0.34 | 0.44 | 0.52 | 0.48 |
| 80_min_busd_log | 0.27 | 0.49 | 0.57 | 0.63 | 0.66 | 0.67 | 0.16 | 0.36 | 0.5 | 0.68 | 0.86 | 1 | 0.094 | 0.22 | 0.31 | 0.41 | 0.51 | 0.59 |
| 1_min_busd_time | 0.46 | 0.22 | 0.16 | 0.12 | 0.096 | 0.084 | 0.11 | 0.15 | 0.14 | 0.12 | 0.11 | 0.094 | 1 | 0.45 | 0.32 | 0.24 | 0.18 | 0.15 |
| 5_min_busd_time | 0.27 | 0.52 | 0.4 | 0.31 | 0.25 | 0.23 | 0.043 | 0.24 | 0.28 | 0.27 | 0.25 | 0.22 | 0.45 | 1 | 0.72 | 0.54 | 0.43 | 0.36 |
| 10_min_busd_time | 0.23 | 0.47 | 0.55 | 0.44 | 0.36 | 0.32 | 0.036 | 0.18 | 0.32 | 0.36 | 0.34 | 0.31 | 0.32 | 0.72 | 1 | 0.75 | 0.59 | 0.51 |
| 20_min_busd_time | 0.22 | 0.44 | 0.53 | 0.59 | 0.5 | 0.44 | 0.035 | 0.16 | 0.26 | 0.42 | 0.44 | 0.41 | 0.24 | 0.54 | 0.75 | 1 | 0.79 | 0.68 |
| 40_min_busd_time | 0.21 | 0.44 | 0.53 | 0.6 | 0.63 | 0.57 | 0.038 | 0.15 | 0.24 | 0.37 | 0.52 | 0.51 | 0.18 | 0.43 | 0.59 | 0.79 | 1 | 0.85 |
| 80_min_busd_time | 0.21 | 0.45 | 0.55 | 0.61 | 0.65 | 0.67 | 0.04 | 0.15 | 0.24 | 0.35 | 0.48 | 0.59 | 0.15 | 0.36 | 0.51 | 0.68 | 0.85 | 1 |

Figure 34 - heatmap of various busd features

Figure 35 shows the heatmaps of correlations between the buying up / selling down features and the x-minute return responses. There is perhaps some relationship at the 80-minute busd variables and the 80-minute returns; these are positively correlated. There may be other relationships too, but this is not entirely clear.

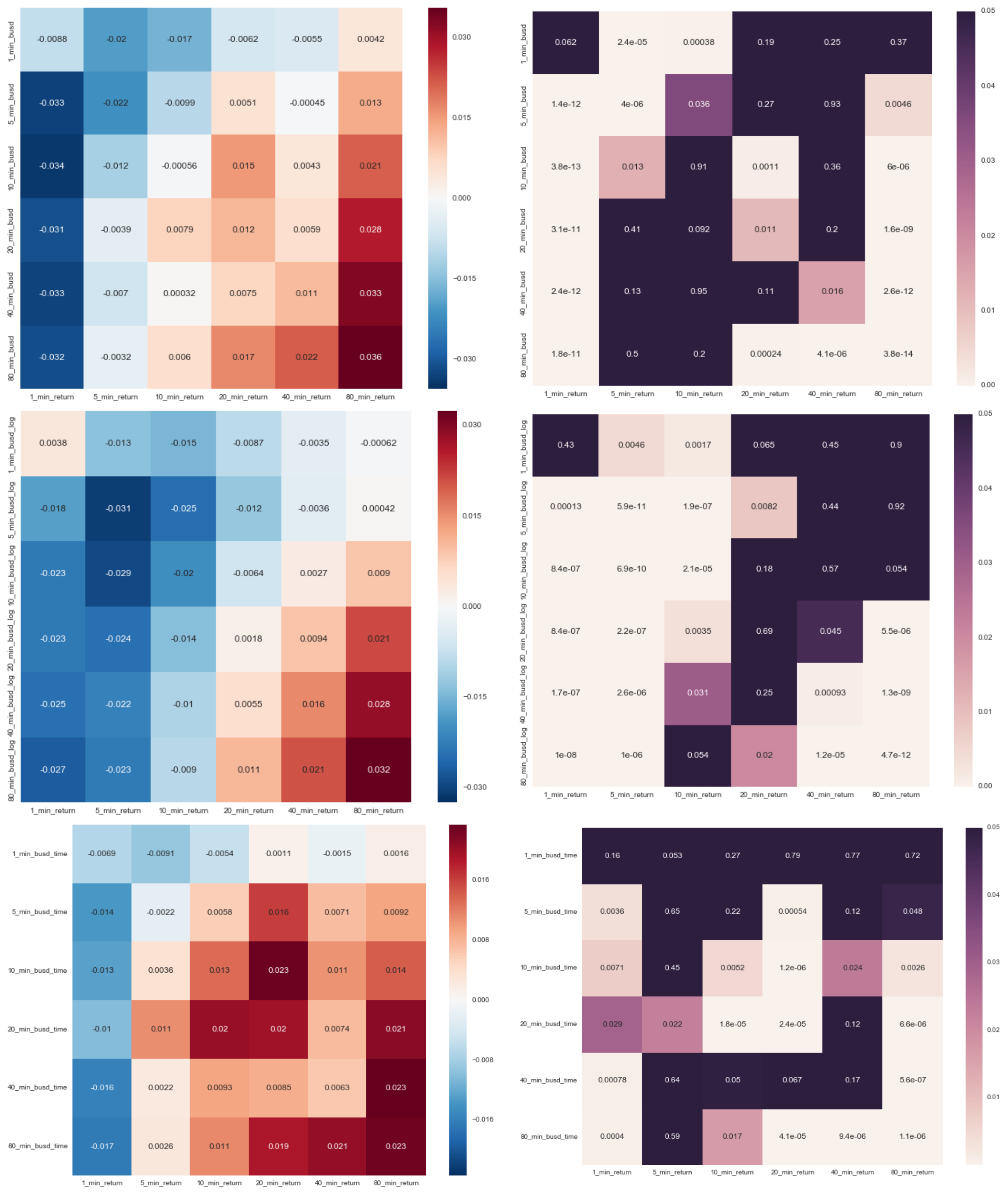


Figure 35 - busd features vs x-minute return responses

Finally, we look at the price vs. high-low range, vs returns. Figure 36 and Figure 37 show there is a positive correlation between longer term price vs range values and longer term returns. Figure 38 nad Figure 39 try to visualize this relationship, but the magnitude of the association is so small that it is not obvious. (Figure 39 shows the distribution of returns when price is above the range, in the upper half of the range, in the lower half of the range, and below the range of the last 80-min high-low respectively.)

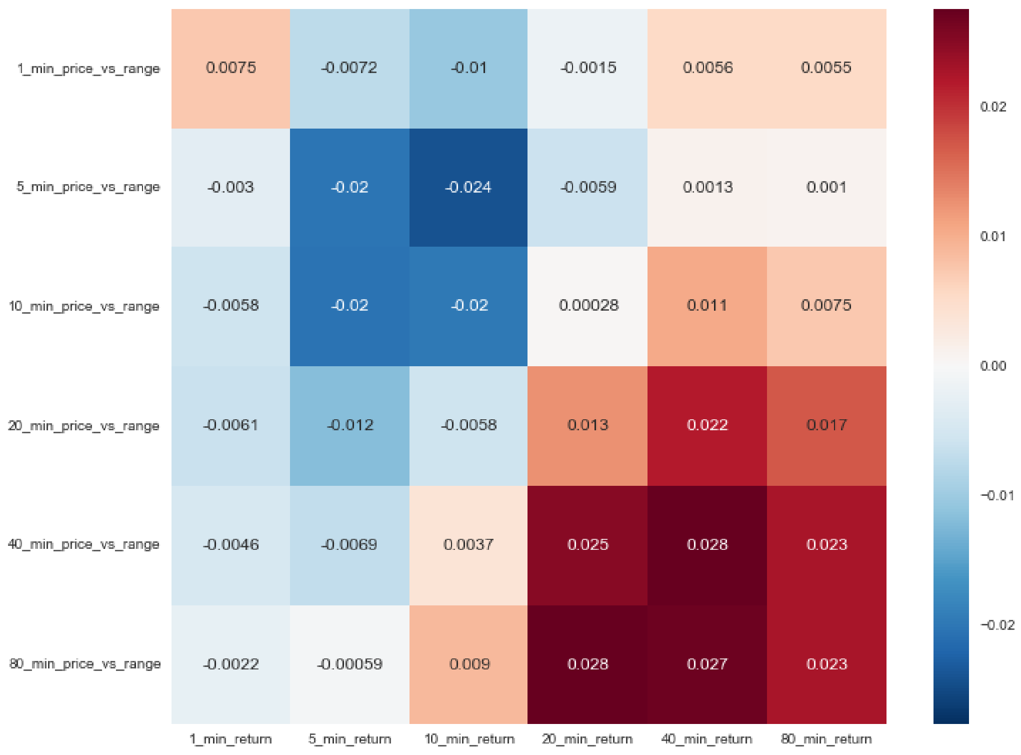


Figure 36 - price vs range vs x-minute return correlation heatmap

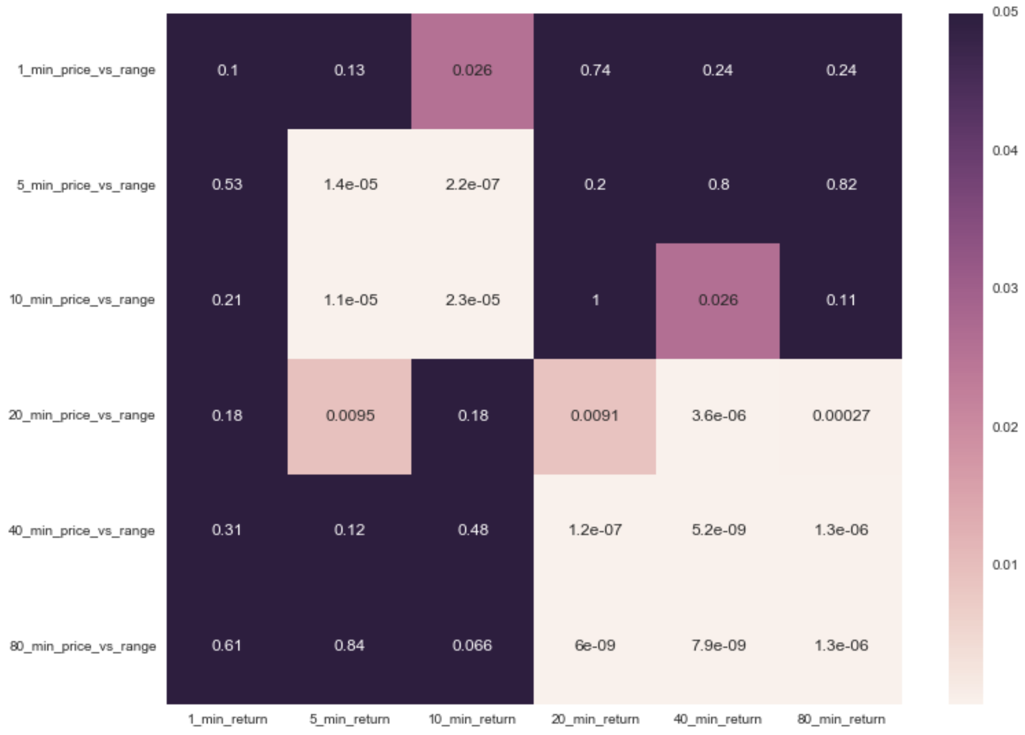


Figure 37 - price vs range vs x-minute return correlation p-value heatmap

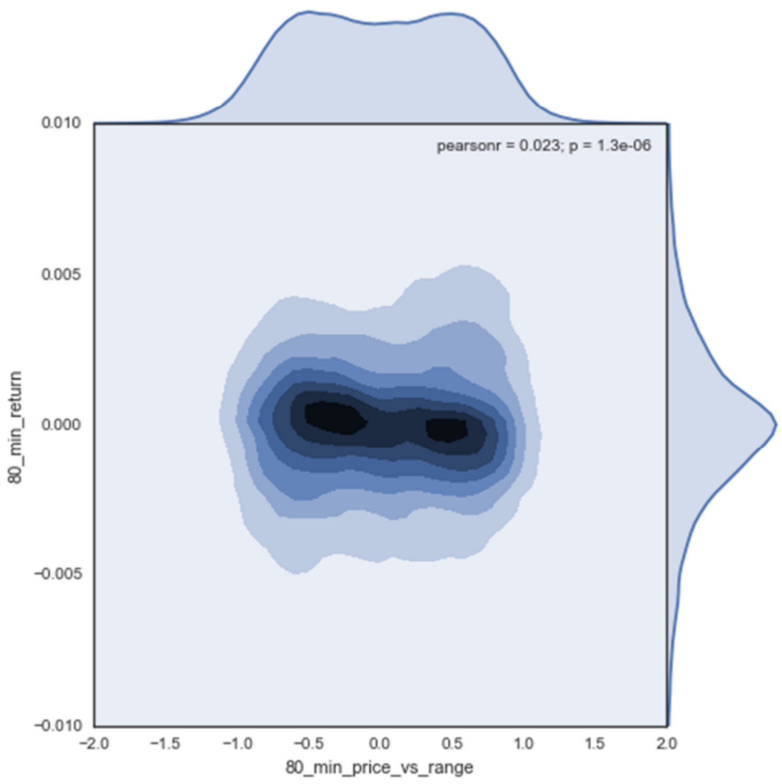


Figure 38 - 80-minute price vs high-low range vs 80-minute return kde plot

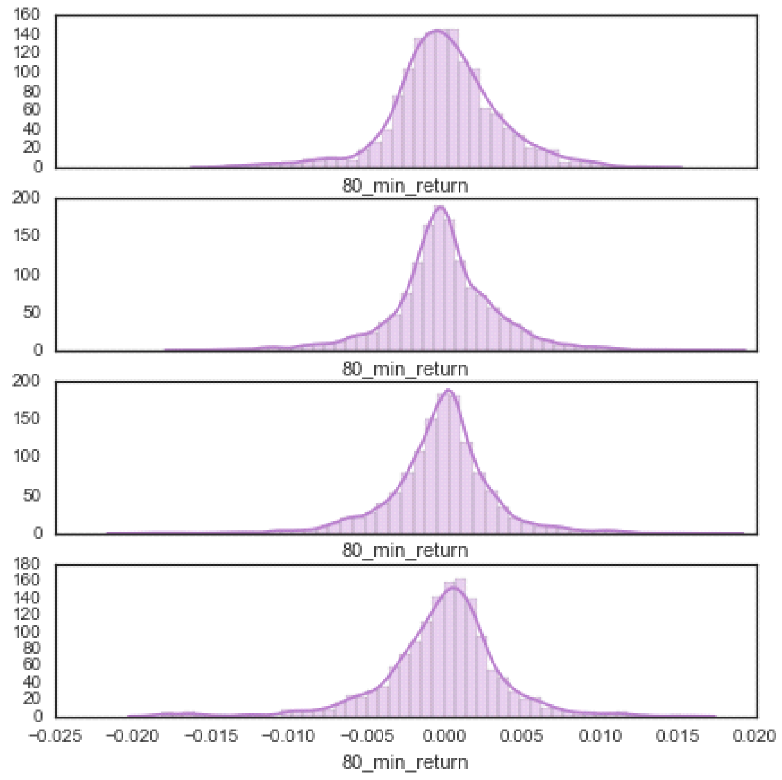


Figure 39 - 80-minute return distribution striated by price vs 80-minute high-low range values

Cross validation strategy

As we are working with time series data, there is a good case to be made that the ordering in the data is important. In particular, we want to avoid 'look ahead' bias. This is because when we do use the model for real world trading, it can only possibly know about things that have happened in the past, and will be called upon to make predictions about the future. Thus, it may not make sense to have the model train on data that is chronologically later, and make predictions about chronologically earlier data.

Thus, our n-fold cross validation strategy is to divide the data into n+1 chronologically contiguous folds. In the first case we have the earliest fold be the training data, and the remaining n folds the test data. In the second case we have the earliest 2 folds be training data, and the remaining n-1 folds test data. In the last case we have the earliest n folds be the training data, and the last fold the test data. This is illustrated in Figure 40

| | | | | | |
|---|---|---|---|---|---|
| Train | Test | Test | Test | Test | Test |
| Train | Train | Test | Test | Test | Test |
| | | | … | | |
| Train | Train | Train | Train | Train | Test |

Figure 40 - Cross-validation strategy

Dimensionality reduction

We use PCA to attempt a dimensionality reduction on the data. Starting with 43 feature columns (Day of Week, Hour, Prior return (x6), last vs MA (x5), log volume vs hx (x6), busd log (x6), busd time (x6), high-low (x6), and price vs range (x6)), we try PCA on 3 folds of the data just to see how stable the results are. This is not a proper cross-validation, in that we are not looking to see how the PCA 'performs' on a a test set; rather we just want to check that the PCA results are reasonably consistent on each fold.

Figure 41 shows the elbow plots of explained variance by number of principal components. These look similar across all three folds. Figure 42 attempts to visualize the first 10 eigenvectors for each fold (the first few eigenvectors do indeed seem similar across folds, though the definition of positive and negative values is sometimes different across folds). Overall, it's not entirely clear where a 'sharp' cutoff encapsulating most of the variance would be. Looking at the total variance explained – we could pick the first 8 eigenvectors and thus retain 80% of the variance, or the first 14 eigenvectors for 90% of the variance. Both of these represent a substantial reduction in the dimensionality of the problem

Figure 43 shows the elbow plot for the PCA on the entire dataset. Figure 44 shows the eignevectors for the PCA on the entire dataset. These are both consistent with the individual folds we tested earlier. We save the PCA transformed variables as pca_v0, pca_v1, … , pca_v42.

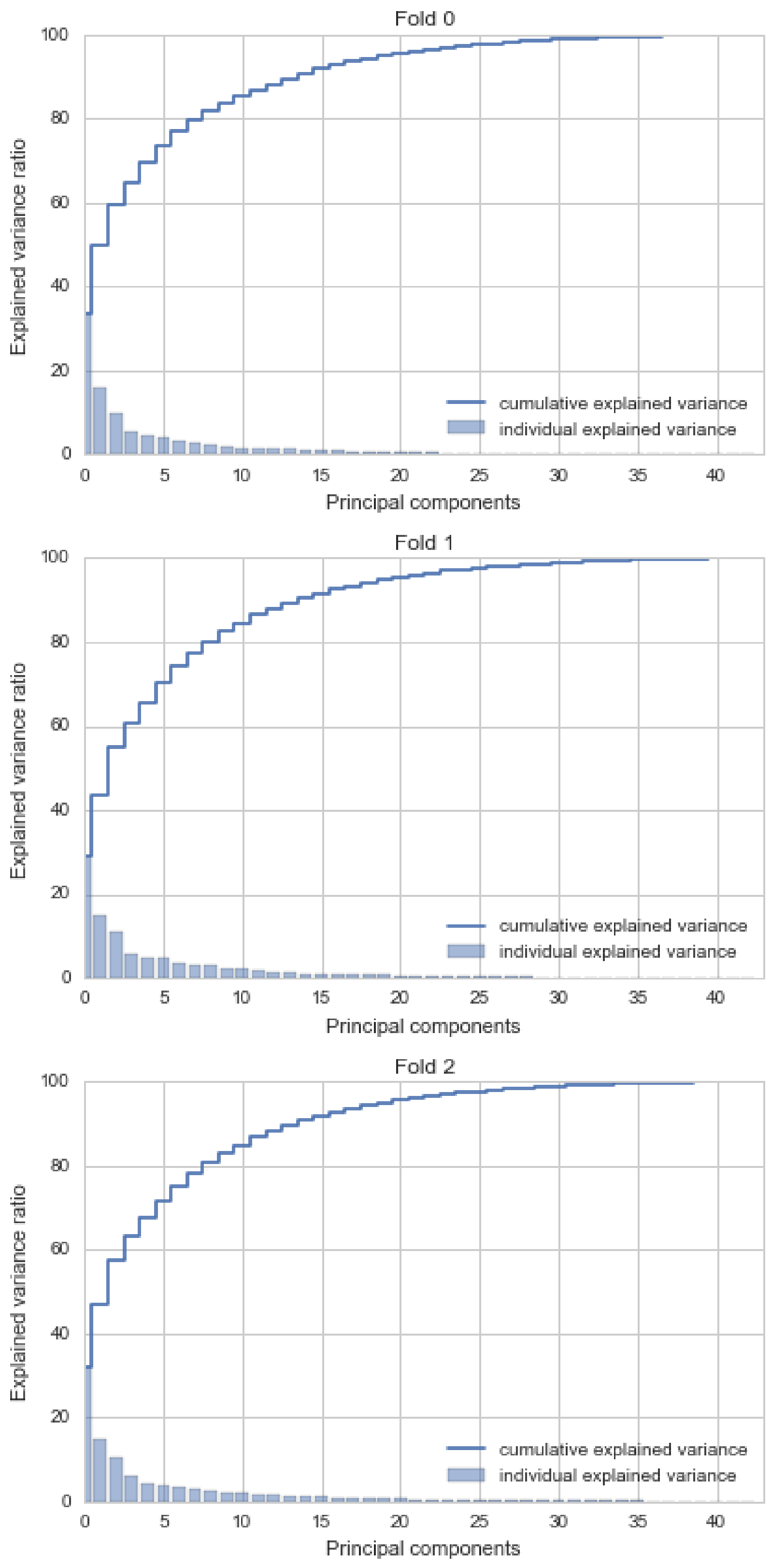


Figure 41 - PCA elbow plots

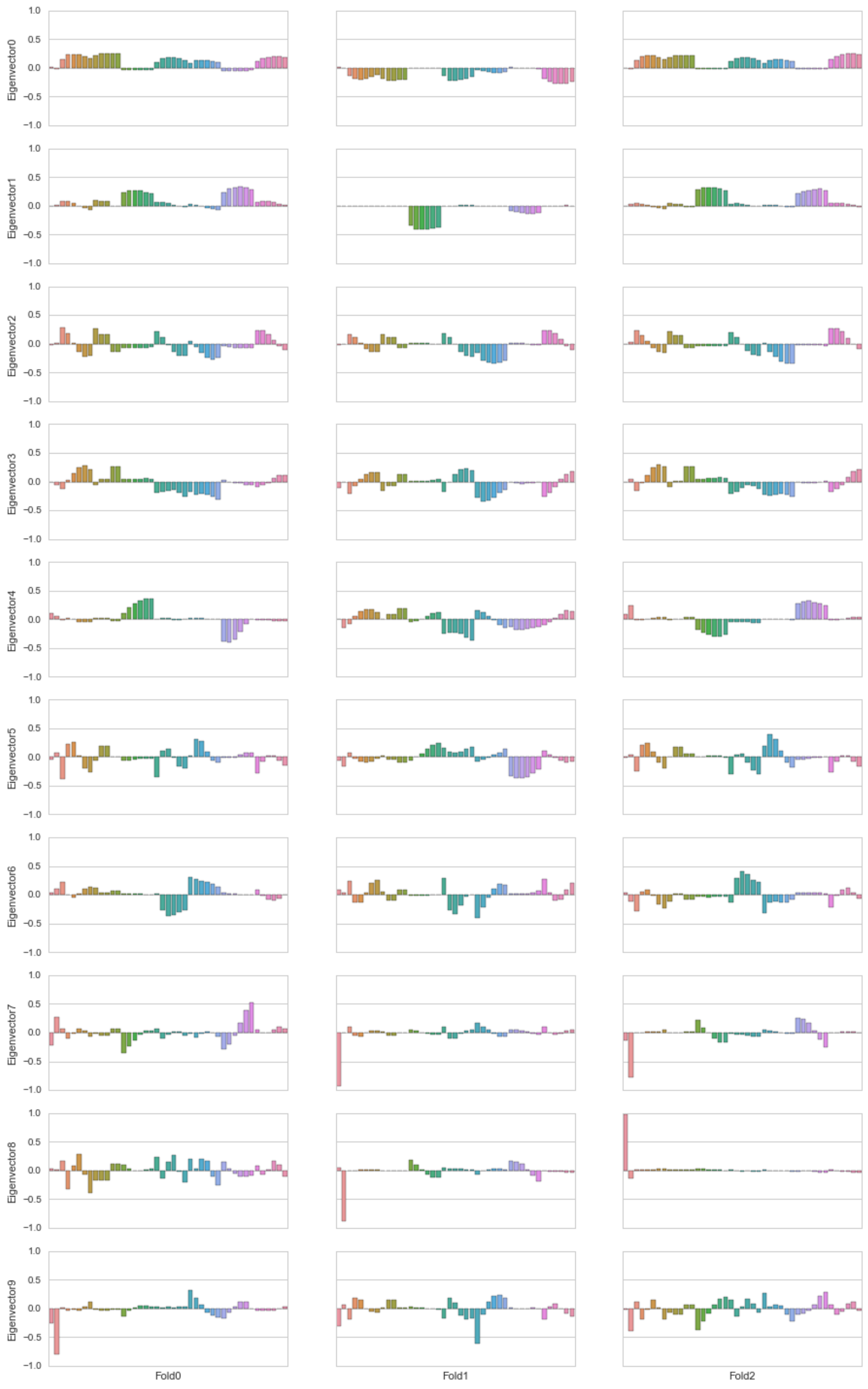


Figure 42 - PCA eigenvectors by fold

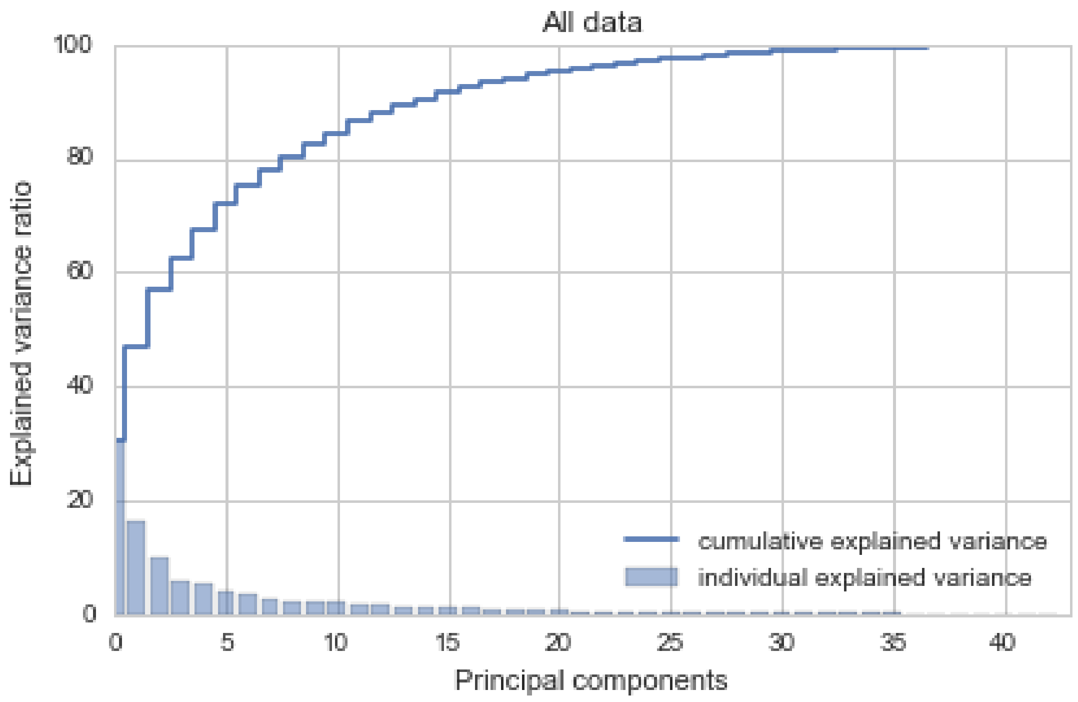


Figure 43 - Elbow plot for PCA on entire dataset

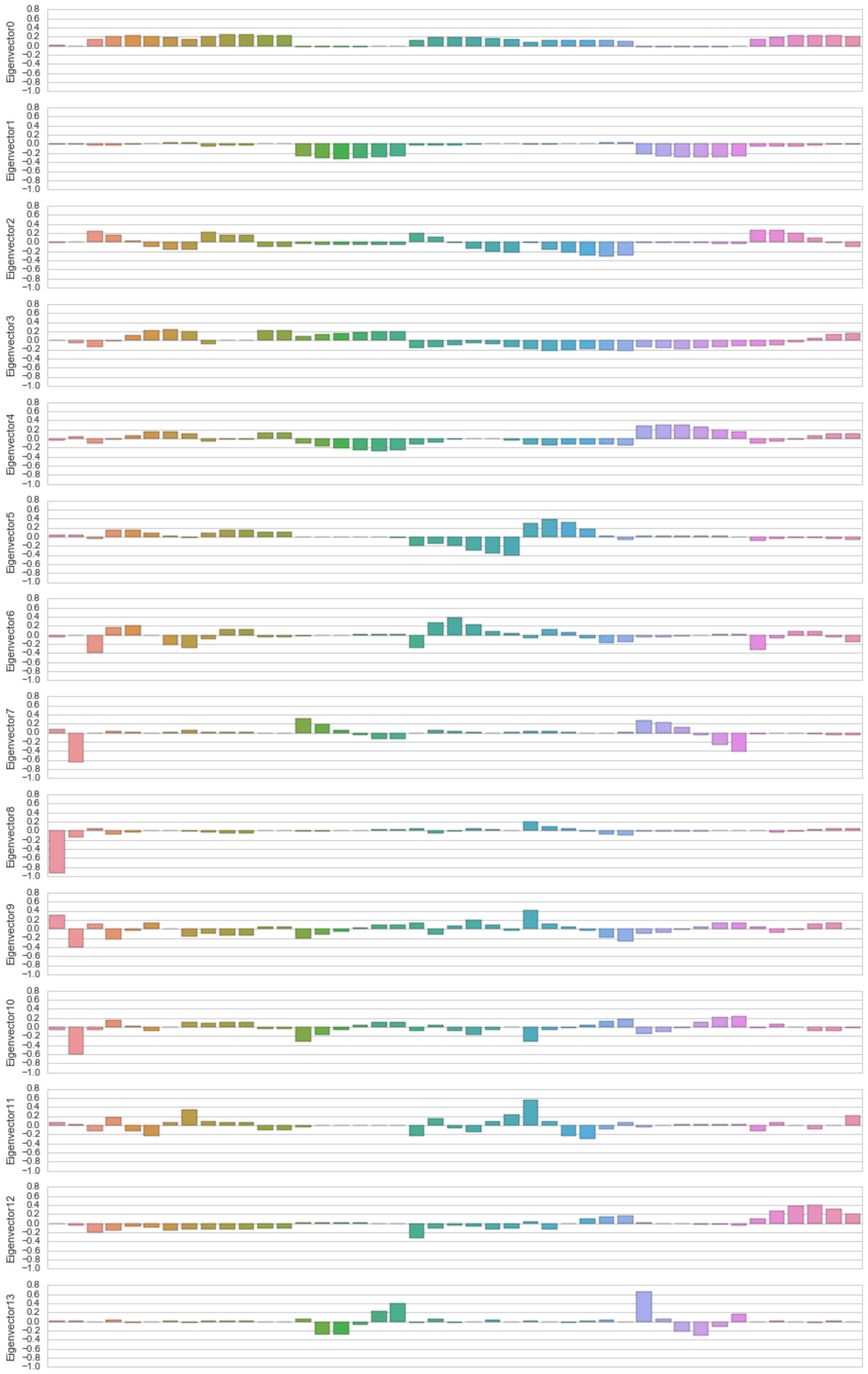


**Figure 44 - Eigenvectors for PCA on entire dataset**

## Decision Tree Analysis

First we attempt to fit decision tree models to the data. We'll fit a regression tree to the x-minute-return responses; we'll also fit classification trees to the x-minute up/down and x-minute categorical responses.

Before selecting a model, we'll choose the maximum depth of the tree and the number of features to use via cross-validation, as described earlier. We'll use 4 folds for cross-validation. The search space is several selected choices between 2 and 15 features, and selected options between a maximum tree depth of 1 and 20.

Table 9 through Table 12 show the best selected hyperparameters for the regression trees and classification trees. Each of these tables shows the best setup for each response, and the cross-validation metric score. When looking at the regression tree and assessing by RMSE, it seems like our models are not particularly predictive, and the best models have a tree depth of 1, suggesting the model is not helping much. But, when we try classification trees on the binary and categorical data, the best models are richer, with higher numbers of features and more depth. In the case of categorical data, we observe that the baseline would be 1/3 or about 33.3%, but in fact we are able to garner a 37% or higher accuracy, which is promising given the difficulty of the problem. Finally, we try examining what scoring on trade Z would yield. Trade Z takes the 'trade recommendation' – i.e., if the predicted return is > 0, 'buy', if < 0, 'sell', if 0, 'do nothing' – and converts that into a trade pnl. So, if the predicted x-minute-return is > 0, and the x-minute-return is 0.05%, the trade pnl is 0.05%. If the predicted return is < 0, and the x-minute return is 0.10%, the trade pnl is -0.1%. If the predicted return is < 0 and the x-minute return is -0.08%, the trade pnl is 0.08%. Finally, we take the mean of these trade pnls and divide y the standard deviation of these trade pnls, to obtain a score akin to a Sharpe ratio. We'd want higher scores on this. Table 12 shows that this suggests richer models. The trade Z's are however somewhat low.

| response | num_features | max_depth | RMSE |
|---|---|---|---|
| 1_min_return | 4 | 5 | 0.000387132 |
| 5_min_return | 2 | 1 | 0.000836649 |
| 10_min_return | 2 | 3 | 0.001165865 |
| 20_min_return | 2 | 1 | 0.001641784 |
| 40_min_return | 8 | 1 | 0.002350687 |
| 80_min_return | 3 | 1 | 0.003321629 |

**Table 9 - Hyperparameter selection for regression tree**

| response | num_features | max_depth | accuracy |
|---|---|---|---|
| 1_min_updown | 4 | 2 | 0.515524113 |
| 5_min_updown | 4 | 4 | 0.521898772 |
| 10_min_updown | 4 | 2 | 0.522039284 |
| 20_min_updown | 4 | 1 | 0.513843936 |
| 40_min_updown | 8 | 3 | 0.508172264 |
| 80_min_updown | 4 | 6 | 0.512490034 |

**Table 10 - Hyperparameter selection for classification tree on binary responses**

| response | num_features | max_depth | accuracy |
|---|---|---|---|
| 1_min_cat | 15 | 6 | 0.398028584 |

| | | | |
|---|---|---|---|
| 5_min_cat | 15 | 4 | 0.397891967 |
| 10_min_cat | 15 | 3 | 0.37814875 |
| 20_min_cat | 8 | 5 | 0.371417542 |
| 40_min_cat | 8 | 4 | 0.370555819 |
| 80_min_cat | 15 | 5 | 0.372679473 |

Table 11 - Hyperparameter selection for classification tree on categorical responses

| response | num_features | max_depth | trade_z |
|---|---|---|---|
| 1_min_return | 8 | 18 | 0.008236401 |
| 5_min_return | 15 | 8 | 0.017853641 |
| 10_min_return | 15 | 4 | 0.02668648 |
| 20_min_return | 8 | 3 | 0.01954195 |
| 40_min_return | 15 | 3 | 0.080407601 |
| 80_min_return | 15 | 3 | 0.037022152 |

Table 12 - Hyperparameter selection for regression tree using trade Z

Columns to drop:
1_min_ma (same as LAST_PRICE)
1_last_vs_ma
1_min_volumes (same as VOLUME)