- **Feature 0: average_near_price_per_stock**
    - **DSL expression:** groupby(key=stock_id, value=ask_price, statistic_operator=mean)
    - **Thought:** Considering the data distribution, especially the potential issue of uneven grouping due to a small number of stocks, we can add a new grouping criterion, 'date_id', to the existing feature 'average_ask_price_per_stock'. This will allow for further refinement when calculating the average ask price for each stock, breaking it down by date to reduce the imbalance caused by having a small number of stocks in each group. Specifically, the implementation will involve using both 'stock_id' and 'date_id' as grouping criteria in the 'groupby' operation.

**Note:** Each feature's *thought* originates from an improvement suggestion made by the LLM after analyzing a corresponding subtree in the syntax tree. After all proposed improvements are implemented and evaluated, the one that performs best on the validation set is adopted.

Specifically, the syntax tree for feature 0 is:

```
└── groupby
  ├── keys
  │   └── stock_id
  ├── value
  │   └── ask_price
  └── statistic_operator
    └── mean
```

The adopted improvement (i.e., the above *thought*) comes from the keys subtree.
Other subtrees also had proposed improvement suggestions, including:

**value**:
*To gain a complementary perspective on market dynamics, we propose replacing the feature 'ask_price' with 'bid_price' in the value field of the groupby operation. This change enables the analysis to focus on buyer-side pricing behavior, which can be particularly insightful in assessing demand-side pressure and liquidity conditions.*

**statistic_operator**:
*Instead of using the mean, which can be sensitive to outliers, we suggest using the median as the statistical operator. The median provides a more robust measure of central tendency, especially in the presence of skewed distributions or extreme price values, thereby improving the reliability of summary statistics in volatile trading environments.*

**groupby**:
*To improve comparability across different groups and mitigate scale differences in price levels, we suggest incorporating a normalization step within or after the groupby operation. This can involve min-max scaling or z-score standardization applied to the aggregated values.*

**Feature 1: average_ask_price_per_stock_per_date**

- **DSL expression:** groupby(key=[stock_id, date_id], value=ask_price, statistic_operator=mean)

- **Thought:** We can further reduce the noise in the results by introducing a moving average to smooth the ask price. Specifically, when calculating 'average_ask_price_per_stock_per_date', we will first apply a moving average to the near_price and then compute the mean of that.

**Feature 2: average_ask_price_per_stock_per_date_smoothed**

- **DSL expression:** groupby(key=[stock_id, date_id], value=moving_average(ask_price, window_size=5), statistic_operator=mean)
- **Thought:** Given that the current feature 'average_ask_price_per_stock' is calculated based on the mean of the ask price, we could attempt to introduce a combination of price-related features to enhance the diversity and robustness of the feature. Based on the data distribution, it is recommended to replace ask_price with a combination of bid_price and ask_price, calculating their difference. This would provide a more comprehensive reflection of market price dynamics.

**Feature 3: average_ask_bid_price_per_stock_per_date_smoothed**

- **DSL expression:** groupby(key=[stock_id, date_id], value=moving_average(ask_price-bid_price, window_size=5), statistic_operator=mean)
- **Thought:** Since the calculation of the bid-ask spread may not be significantly affected by extreme values, we can introduce additional statistical measures to more comprehensively reflect the characteristics of the spread. Specifically, when calculating the bid-ask spread, in addition to using the mean, we can also introduce quartiles (Q1 and Q3) to capture the distribution characteristics of the spread. We can calculate the quartiles (Q1 and Q3) of the ask_price and bid_price for each stock on each date and use Q3 - Q1 to represent the range of changes in the spread.

**Feature 4: iqr_ask_bid_price_per_stock_per_date_smoothed**

- **DSL expression:** groupby(key=[stock_id, date_id], value=(ask_price - bid_price), statistic_operator=IQR), where IQR represents the interquartile range.
- **Thought:** In order to capture more distribution patterns, we can attempt additional price-related feature combinations such as near_price, wap, far_price, etc.

**Feature 5: iqr_prices_per_stock_per_date_smoothed**

- **DSL expression:** groupby(key=[stock_id, date_id], value=price for price in [near_price, wap, far_price], statistic_operator=IQR)