Homework 1 Write-up

Contributors: Ziheng Chen, Cong Wang, Chengyang Gu

Preparation for TAQ Data

We download and unzip the TAQ data from Google drive. The data contains all quotes and trades, in the format of gzip streams, of over 8,000 stocks on NYSE on all trading days from June 20 to September 20, 2007. Each file corresponds to a stock on a particular day. The number of files is not consistent on different days, presumably because our data provider do not include stocks with no orders on any day. The data takes about 17 GB of storage.

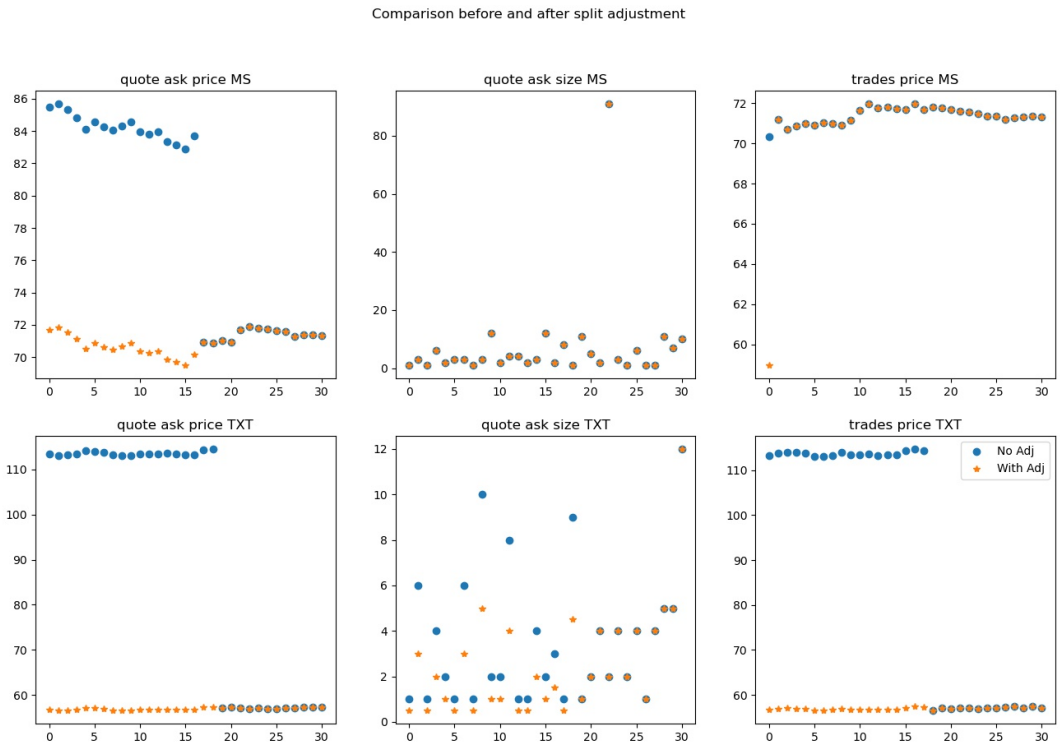
⚠️ Caveat: To effectively move the data, compress them first. Fully decompressed data is distributed in over 1 million small files and due to hardware concerns, it can take days to relocate if not compressed first.

We use `s_p500.xlsx`, shipped with the data, to filter out stocks that are not in the S&P 500 during the time period of interest. There are two stocks that need special treatment. The first is "SUNW", which changed its symbol to "JAVA" on August 27. We identify their sameness by their CUSIP. We use "JAVA" throughout our analysis. The second is "CMCSA", whose data is not available by our data provider. We simply ignore this stock.

Split-adjustment

We create a script to adjust for events where stock prices or volumes jump because of split. The split adjustment is performed using the cumulative factors in `s_p500.xlsx`. Our code is able to automatically analyze all the stocks, adjust accordingly, and save the adjusted data on demand. It also makes a simplified assumption that each stock only experience at most one split in the period of interest, which is proven by a unit test. Such assumptions that particularly apply to the case of concern are common throughout our project because they streamline development.

Based on the split adjusted data, we plot graphs of quote ask prices, sizes and trade prices before and after adjustment for stock MS and TXT, in order to prove our code work as expected. Both stocks experience split in the period. We plot the data one day before and on the split day. The points are chosen at a fixed interval for each series. Blue points correspond to the series without adjustment, while orange stars correspond to those after adjustment. Ideally, after split the adjusted data before and after adjustment should coincide, while the prices before and after splits after adjustment should be close. We do observe this in the graph below. This testifies out intuition that split does not affect the value of the stock.



Cleaning

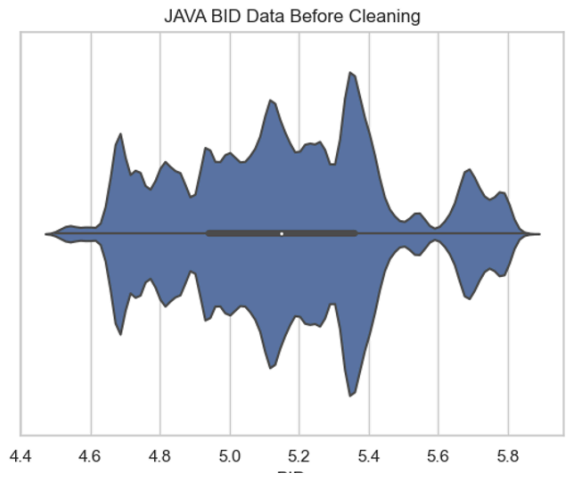
There are jumps and outliers in this dataset. Jumps are defined as correct data points with uncommon volatilities and outliers are just wrong data points. Some model may allow the existence of jumps, while some models are sensitive to outliers, such as the impact model. For different ways of using this dataset, we may need to clean this dataset with different ways. So, we have two kinds of cleaning process and users can choose which method to use according to their needs.

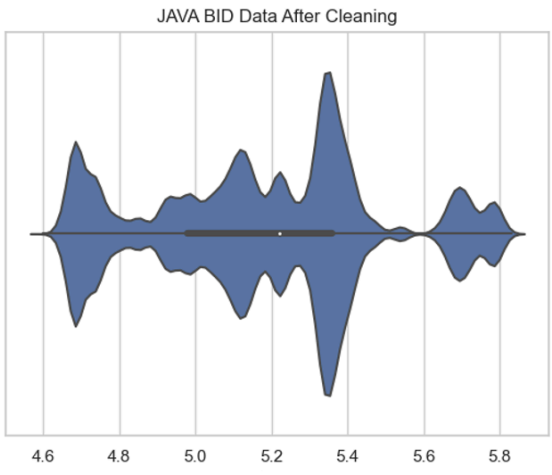
The first method is to use the k-day rolling mean and rolling standard error. For data points outside two errors, they are cleaned. This method mainly focuses on cleaning outliers and may keep the majority of the jumps. This method is more suitable for preparing data for models which need to keep as many correct data as possible and are not sensitive to jumps.

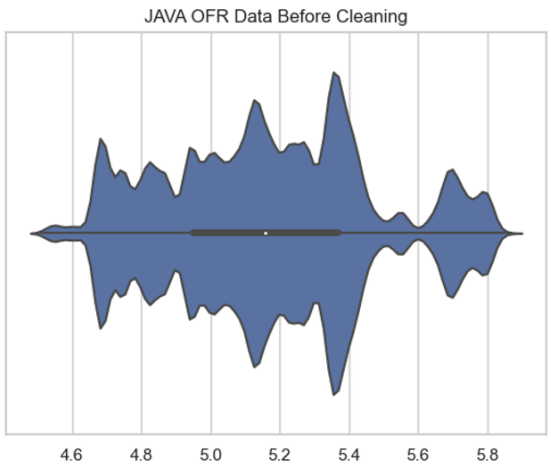
The second method is to use the k continuous tick data rolling mean and standard error. Just like the first method, data points outside two or three errors are cleaned. Choosing whether two or three errors may depend on the model we are going to use. This method can clean both jumps and outliers and is more suitable for preparing for models which need more stable data.

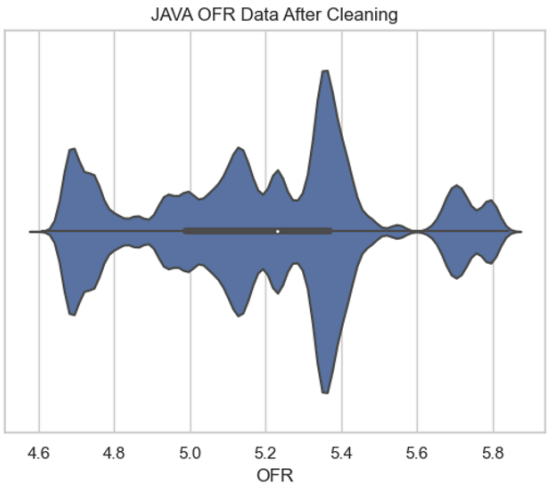
In this assignment, we use the former one because it is more intuitive and easier to explain.

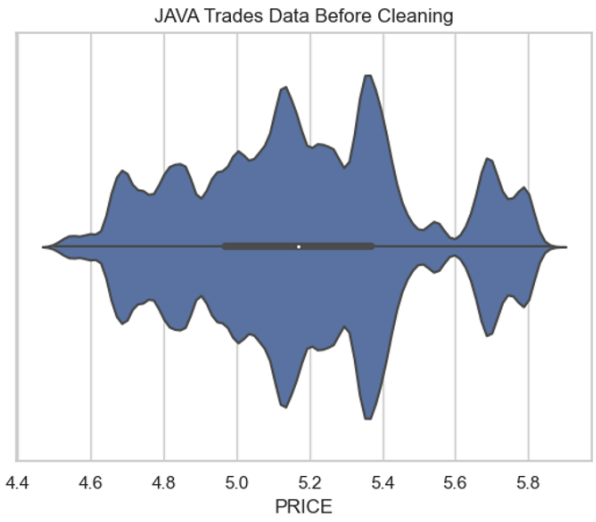
To show the cleaning result, take JAVA for example. We output violin plots wo see the distribution of data before cleaning and cleaned data.

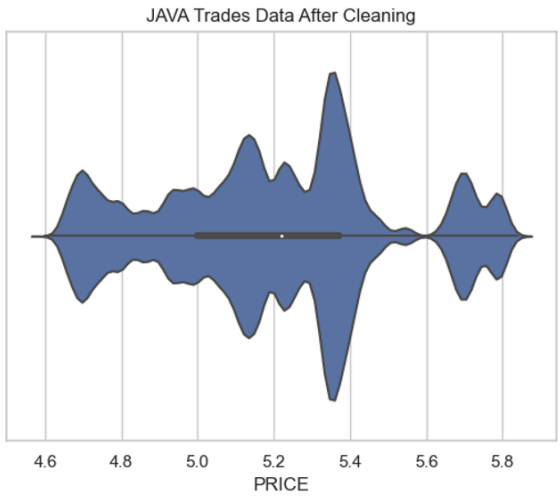






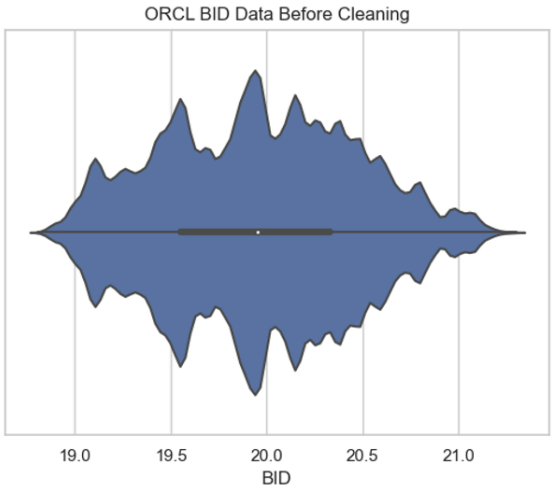


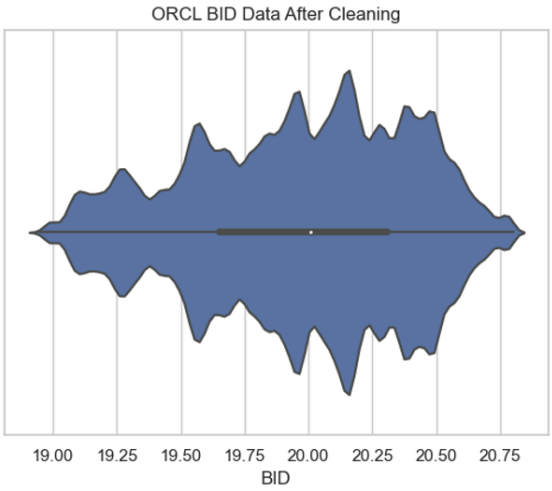


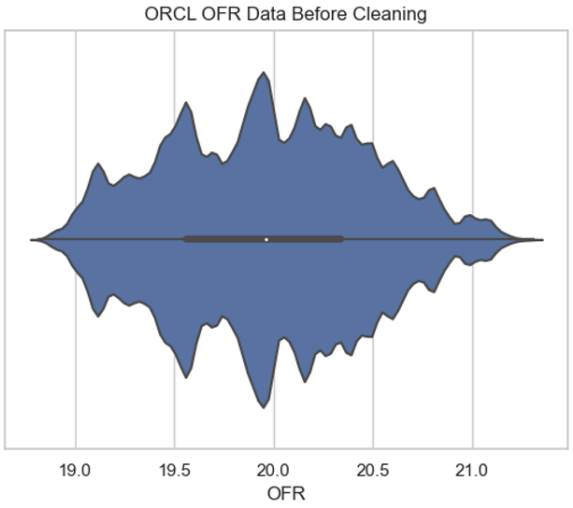


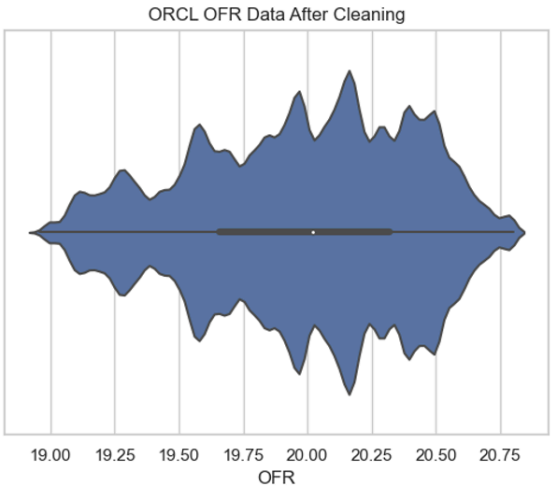
Clearly, we can see that cleaned data has few outliers. We can easily find that there are fewer data points in the tails. There are very few data points fall below 4.6 dollars after cleaning.

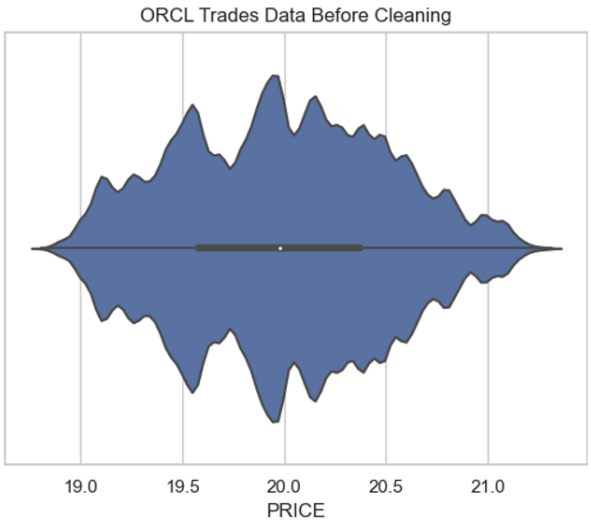
Take ORCL for another example.

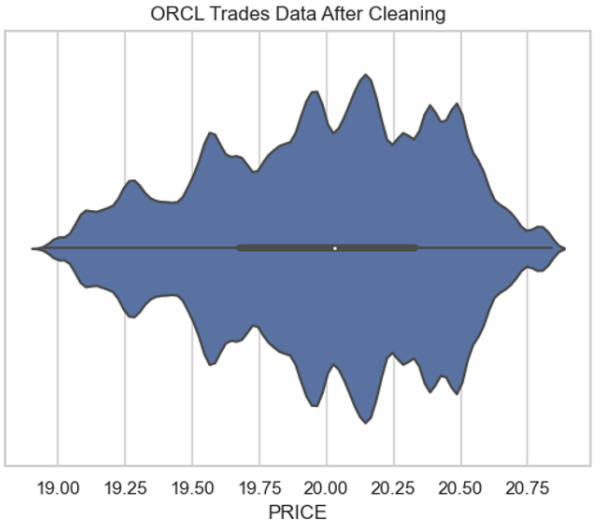












Clearly, we can see that cleaned data has few outliers. We can easily find that there are fewer data points in the tails. There are very few data points higher than 21.0 dollars after cleaning.

Saving

We design two methods, saving data as csv files and saving data as pickle files, to save the cleaned TAQ data. There is a trade-off between reading speed and storage space.

If we save data in the form of csv, we can save space by making some treatment before saving them as csv files. We substitute the duplicated rows with NaN, which does not take storage space. We also use multi-index to save place in the index. Moreover, we keep 2 decimal places instead of 15 decimal spaces. The space after treatment is about one fifth of the space originally needed without any treatment. We need more than 100G to store the cleaned TAQ data but only 20G is needed after these treatments.

We can also choose to save data as pickle files. The advantage of saving files as pickle files is reading speed. Reading data from pickle can be 40 times quicker than reading data from csv files, while it may take three times more storage space to storage these files compared with saving files as csv files.

Users may choose different method of saving cleaned TAQ data according to their needs.

Statistics Summary

Methodology

1. Compute X-minute returns of trade and mid-quotes.

The way we solve this problem is to use the method resample from Pandas package. It allows us to specify a time interval in minute and group price for each time interval with a starting time. For example, with 10 minutes, we start at 9:30 am, and it will group prices into 10 minutes bucket and then we compute the average price within each bucket and label it on the right side. In this case, we would have a series of price at 9:40am, 9:50am, 10:00am, etc. Next, we compute the simple price return of this new series to be our return as $(P_t - P_{t-1})/P_t$.

2. Basic statistics

We use the following formula to annualize price return and 252 trading days is used for a year.

$$AnnualizedReturn = (1 + Return_{Freq})^{NumFreq} - 1$$

Freq is the time interval, e.g., 5 minutes, 30 minutes, etc. NumFreq is the number of Freq in a whole year (252 days). For example, with 5 minutes interval, the NumFreq would be equal to $252 * 24 * 60/5 = 72,576$.

We compute the std of the average price series and multiply by the square root of the number of freq in a whole year.

$$AnnualizedStandardDeviation = StandardDeviation_{Freq} * \sqrt{NumFreq}$$

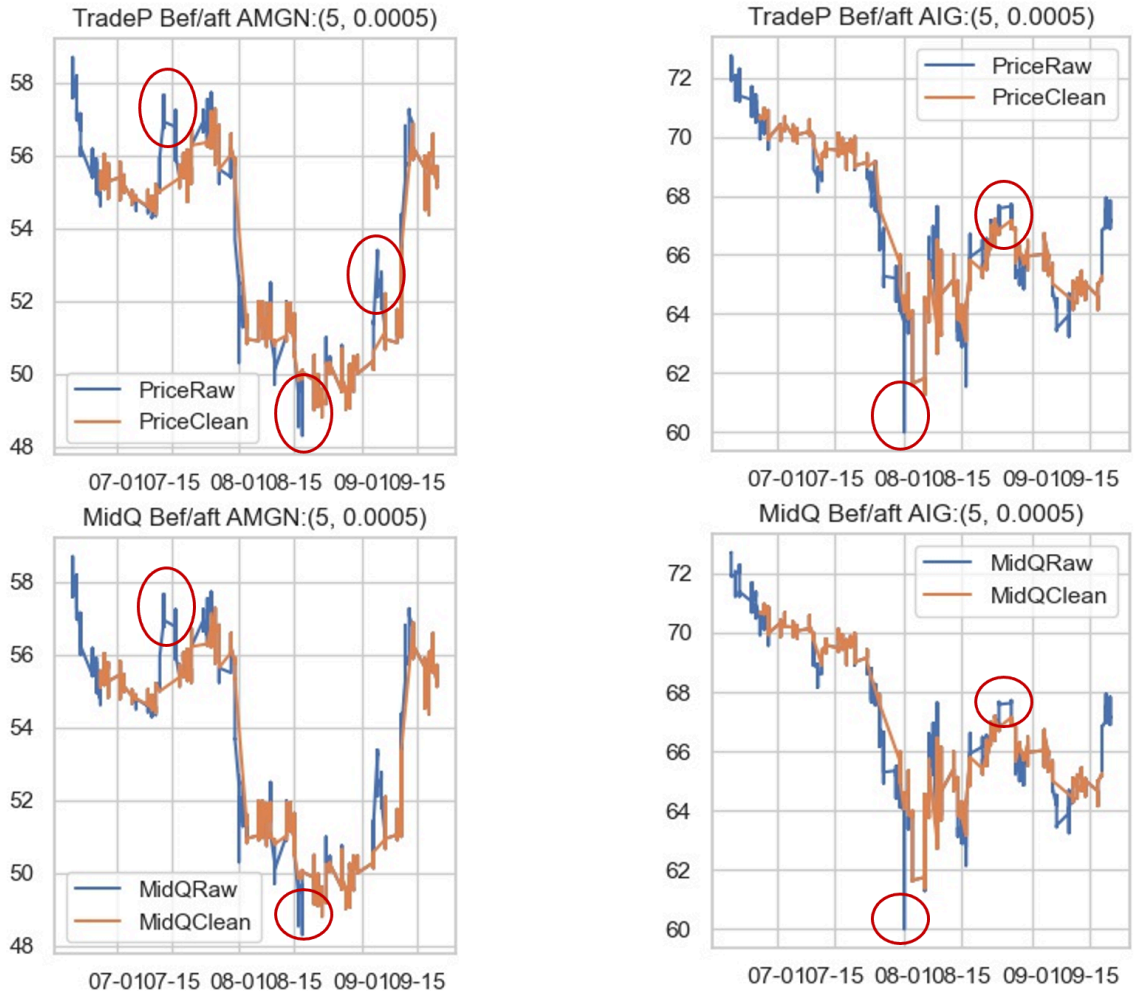
In our code, we use `scipy.stats` to compute the skewness and kurtosis.

For normally distributed data, skewness should be about zero. For unimodal continuous distributions, a skewness value greater than zero means that there is more weight in the right tail of the distribution.

Kurtosis is the fourth central moment divided by the square of the variance. We used Fisher's definition in our code, and thus 3 should be subtracted from the result to give 0 for a normal distribution.

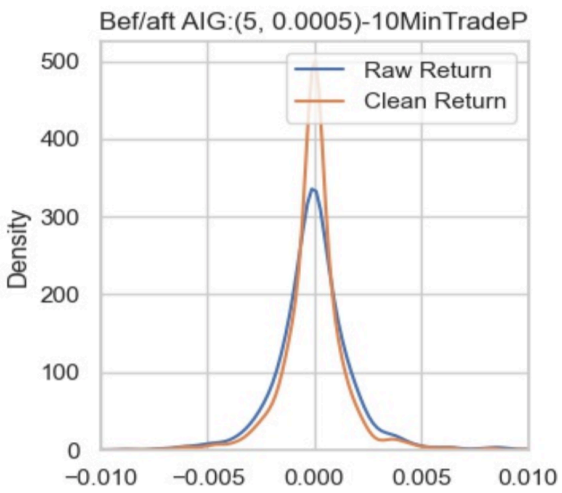
Analyze Cleaning Impact

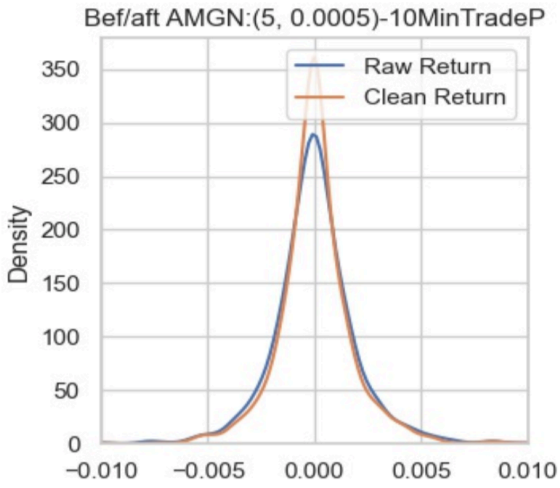
We picked AMGN and AIG to see the impact of using initial cleaning param (5, 0.0005) as below. AMGN is a big and strong biotech company and has experienced a very high growth over the last decade and was relatively stable in Q3 2007. AIG was bailed out by US Government in 2008, so we'd like to see what the price looks like during Q3 2007 which happens to be our sample data.



(Fig.1) Left is for AMGN, right is for AIG. Top half: trade price. Bottom half: mid quote price.

We then plotted price and mid quote data for both cleaned and raw dataset.As we can see from the chart, the cleaned data did remove those outlier data points or jumping moment highlighted in a few red circles. It's smoothed out a little bit compared to the original data. Then we recalculated the basic statistics below and also plot the distribution of the trade price return and mid quote return.





Raw	Clean	stats
65.0	51.0	SampleLen
3263861.0	1616073.0	NumTrades
6728791.0	3286347.0	NumQuotes
0.49	0.49	TtoQ Ratio
-0.45	0.21	mean
-0.32	0.0	median
0.38	0.31	std
0.48	1.9	skew
15.69	23.25	kurt
0.0	0.0	MAD
-1.9	-1.43	MDD

Raw	Clean	stats
65.0	51.0	SampleLen
2985045.0	1676290.0	NumTrades
6305926.0	3644096.0	NumQuotes
0.47	0.46	TtoQ Ratio
0.74	5.01	mean
-0.51	0.0	median
0.39	0.34	std
0.37	0.83	skew
9.84	8.12	kurt
0.0	0.0	MAD
-1.92	-1.59	MDD

(Fig.2)

The charts above is based on 10 min-return from trade price using intial parameters to clean the data. The distribution after cleaning is more clustered to the mean value (not annualized). This is corresponding to our expectation since we removed those outliers or jumps.

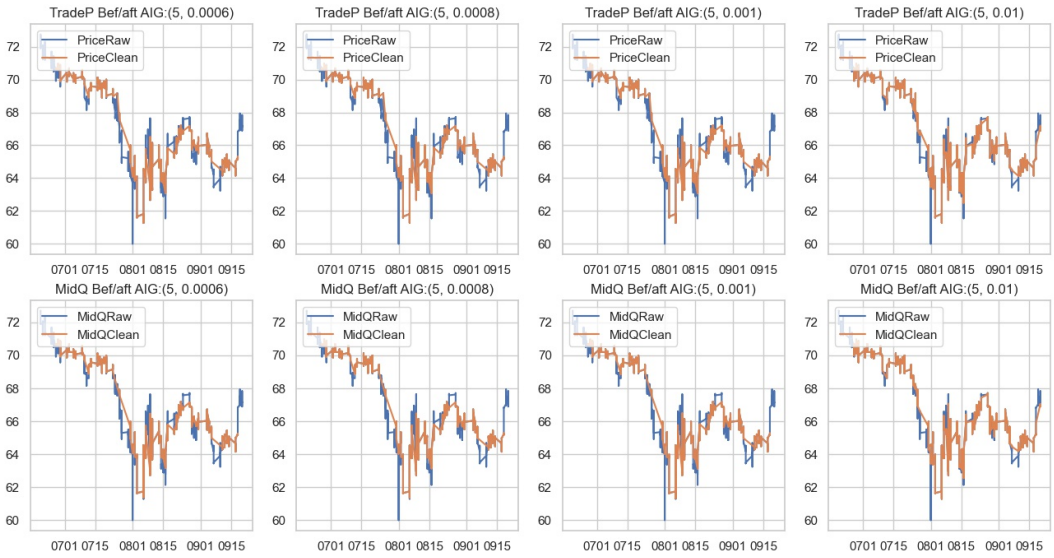
The statistics in the table is annulized value for mean, std, median. The standard deviation is dropped slightly for both stocks after cleaning. Both number of trades and quotes are reduced almost by half, although the trade to quote ratio stays relatively same. The skew increased for AGI after cleaning. Kurtosis stays relatively stable for AMGN but increased for AIG. The absolute value of MaximumDrawDown decreased as we removed some very sharp drop of price. We repeated this for time interval 10 seconds, 30 seconds, 1 minute, 5 minutes, 10 minutes and 30 minutes. Please find all the images in `plot/stats` for both AMGN and AIG with different time interval. Conclusions include:

- Time interval increases, the distribution of the price return tends to be more likely to follow a normal distrition. For example, both skew and kurtosis gradually decrease and getting close to 0. (Kurtosis needs to subtract 3.)
- After cleaning the data, especially for 5, 10, 30-min interval, the price return becomes more

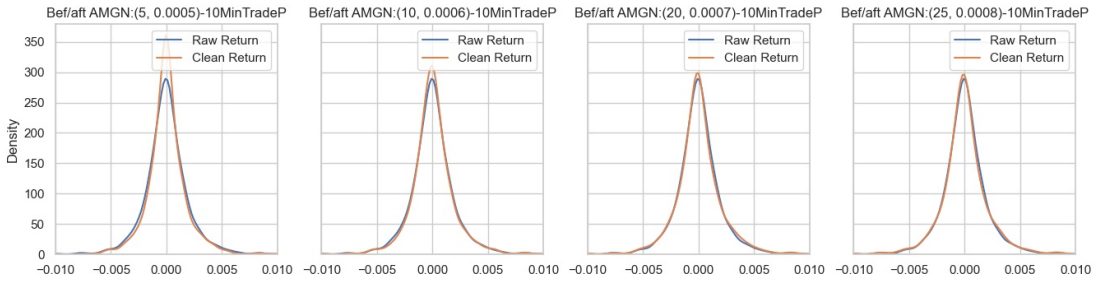
clustered to the mean value as we removed outliers.

- The similar pattern seems to apply to AIG stock as well.

We finally searching for “best” parameters.We would interpret “best” as the parameters that would bring the raw data more close to normal distribution by improving the stats such as kurtosis, skew, std, etc.We noticed that as we increase the parameter k and gamma, it will drop more outliers and data from the dataset. Apparently, we are using past k days to compute the standard deviation of the price. So if we choose k=60, then we will lose majority of the data sample.



The figure below serves as a support to the point mentioned above.



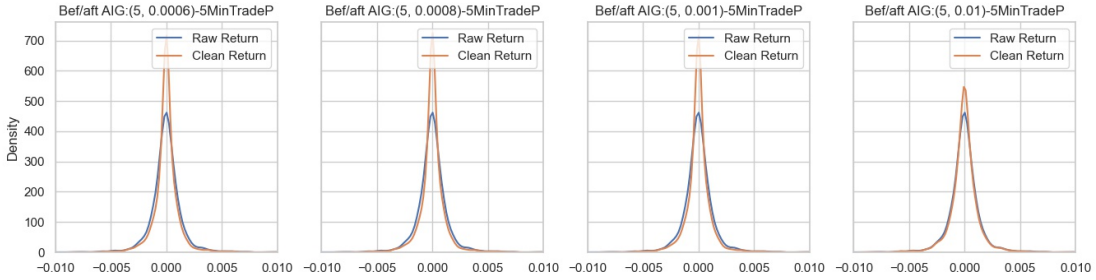
Raw	Clean	stats
65.0	51.0	SampleLen
2985045.0	1676290.0	NumTrades
6305926.0	3644096.0	NumQuotes
0.47	0.46	TtoQ Ratio
0.74	5.0	mean
-0.51	0.0	median
0.39	0.34	std
0.37	0.83	skew
9.84	8.12	kurt
0.0	0.0	MAD
-1.92	-1.59	MDD

Raw	Clean	stats
65.0	45.0	SampleLen
2985045.0	1746375.0	NumTrades
6305926.0	3738680.0	NumQuotes
0.47	0.47	TtoQ Ratio
0.74	6.89	mean
-0.51	-0.44	median
0.39	0.37	std
0.37	0.71	skew
9.84	7.28	kurt
0.0	0.0	MAD
-1.92	-1.58	MDD

Raw	Clean	stats
65.0	39.0	SampleLen
2985045.0	1572518.0	NumTrades
6305926.0	3375441.0	NumQuotes
0.47	0.47	TtoQ Ratio
0.74	2.01	mean
-0.51	0.0	median
0.39	0.4	std
0.37	0.38	skew
9.84	6.65	kurt
0.0	0.0	MAD
-1.92	-1.74	MDD

Raw	Clean	stats
65.0	34.0	SampleLen
2985045.0	1442611.0	NumTrades
6305926.0	3077779.0	NumQuotes
0.47	0.47	TtoQ Ratio
0.74	0.48	mean
-0.51	-0.74	median
0.39	0.4	std
0.37	0.47	skew
9.84	6.75	kurt
0.0	0.0	MAD
-1.92	-1.62	MDD

So, we would like to keep the k up to 25 to keep at least more than half the sample for analysis. Then we keep k constant and explore which gamma may generate better statistics for cleaned data. Also, we kept k unchanged while changing gamma in order to see the impact of it. As we see from AIG (10min) chart below, as we use gamma=0.01, both skewness and kurtosis improved to the best in terms of being close to normal distribution.



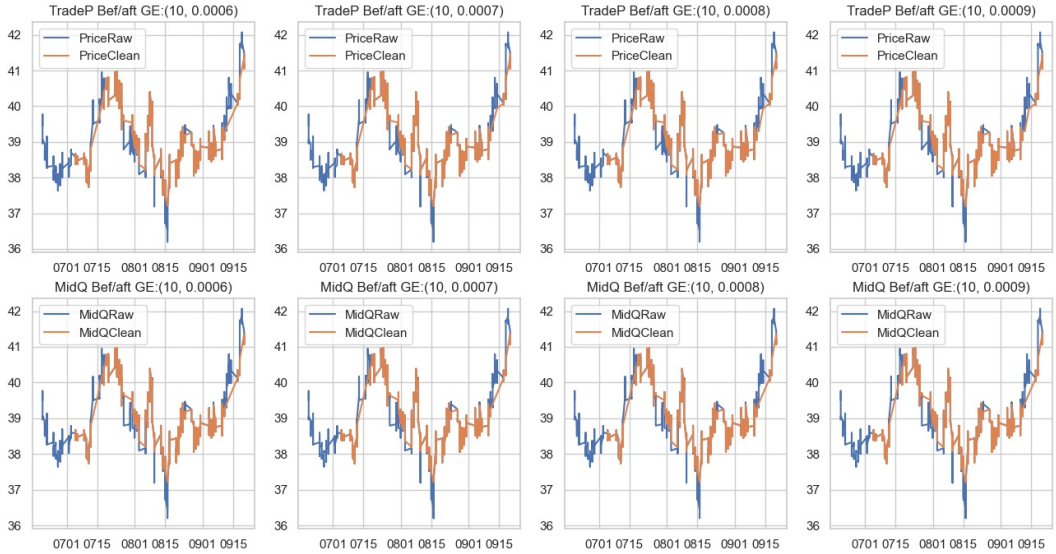
Raw	Clean	stats
65.0	51.0	SampleLen
3263861.0	1623623.0	NumTrades
6728791.0	3301638.0	NumQuotes
0.49	0.49	TtoQ Ratio
-0.51	0.02	mean
-0.38	0.0	median
0.4	0.34	std
1.08	2.78	skew
25.92	49.32	kurt
0.0	0.0	MAD
-1.63	-1.34	MDD

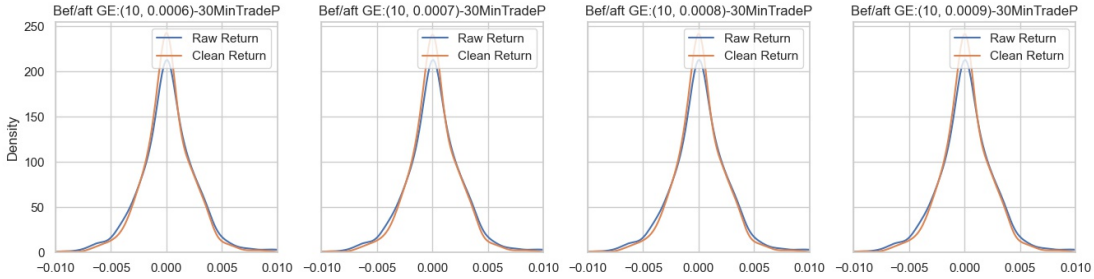
Raw	Clean	stats
65.0	51.0	SampleLen
3263861.0	1644333.0	NumTrades
6728791.0	3334722.0	NumQuotes
0.49	0.49	TtoQ Ratio
-0.51	-0.03	mean
-0.38	0.0	median
0.4	0.34	std
1.08	2.76	skew
25.92	49.27	kurt
0.0	0.0	MAD
-1.63	-1.34	MDD

Raw	Clean	stats
65.0	51.0	SampleLen
3263861.0	1662328.0	NumTrades
6728791.0	3371532.0	NumQuotes
0.49	0.49	TtoQ Ratio
-0.51	-0.1	mean
-0.38	0.0	median
0.4	0.34	std
1.08	2.69	skew
25.92	48.25	kurt
0.0	0.0	MAD
-1.63	-1.34	MDD

Raw	Clean	stats
65.0	57.0	SampleLen
3263861.0	2357757.0	NumTrades
6728791.0	4814979.0	NumQuotes
0.49	0.49	TtoQ Ratio
-0.51	-0.15	mean
-0.38	0.0	median
0.4	0.37	std
1.08	1.85	skew
25.92	31.71	kurt
0.0	0.0	MAD
-1.63	-1.37	MDD

Next, we experimented with some different stocks using trade or quote data to explore the “best” parameter. For example, below is the plot of GE trade and quote data before and after cleaning.





Raw	Clean	stats
65.0	48.0	SampleLen
5872865.0	3571240.0	NumTrades
9937083.0	6152871.0	NumQuotes
0.59	0.58	TtoQ Ratio
4.72	2.45	mean
0.0	0.0	median
0.28	0.24	std
0.04	0.02	skew
3.27	1.65	kurt
0.0	0.0	MAD
-2.29	-2.02	MDD

Raw	Clean	stats
65.0	49.0	SampleLen
5872865.0	3587549.0	NumTrades
9937083.0	6160398.0	NumQuotes
0.59	0.58	TtoQ Ratio
4.72	2.48	mean
0.0	0.0	median
0.28	0.24	std
0.04	0.02	skew
3.27	1.64	kurt
0.0	0.0	MAD
-2.29	-2.02	MDD

Raw	Clean	stats
65.0	50.0	SampleLen
5872865.0	3593118.0	NumTrades
9937083.0	6185023.0	NumQuotes
0.59	0.58	TtoQ Ratio
4.72	2.52	mean
0.0	0.0	median
0.28	0.24	std
0.04	0.02	skew
3.27	1.64	kurt
0.0	0.0	MAD
-2.29	-2.02	MDD

Raw	Clean	stats
65.0	50.0	SampleLen
5872865.0	3596529.0	NumTrades
9937083.0	6199524.0	NumQuotes
0.59	0.58	TtoQ Ratio
4.72	2.57	mean
0.0	0.0	median
0.28	0.24	std
0.04	0.01	skew
3.27	1.64	kurt
0.0	0.0	MAD
-2.29	-2.02	MDD

For GE trade price (30 min), the “best” may be (10, 0.0009) because it does improve skew and kurt while keeping most data points in the dataset. It may be a tradeoff between keeping more data points and removing all the outliers.

We summarize all the findings and pin down the parameters $(k = 20, \gamma = 7 \times 10^{-4})$ for the entire cleaning process and all the analysis below.

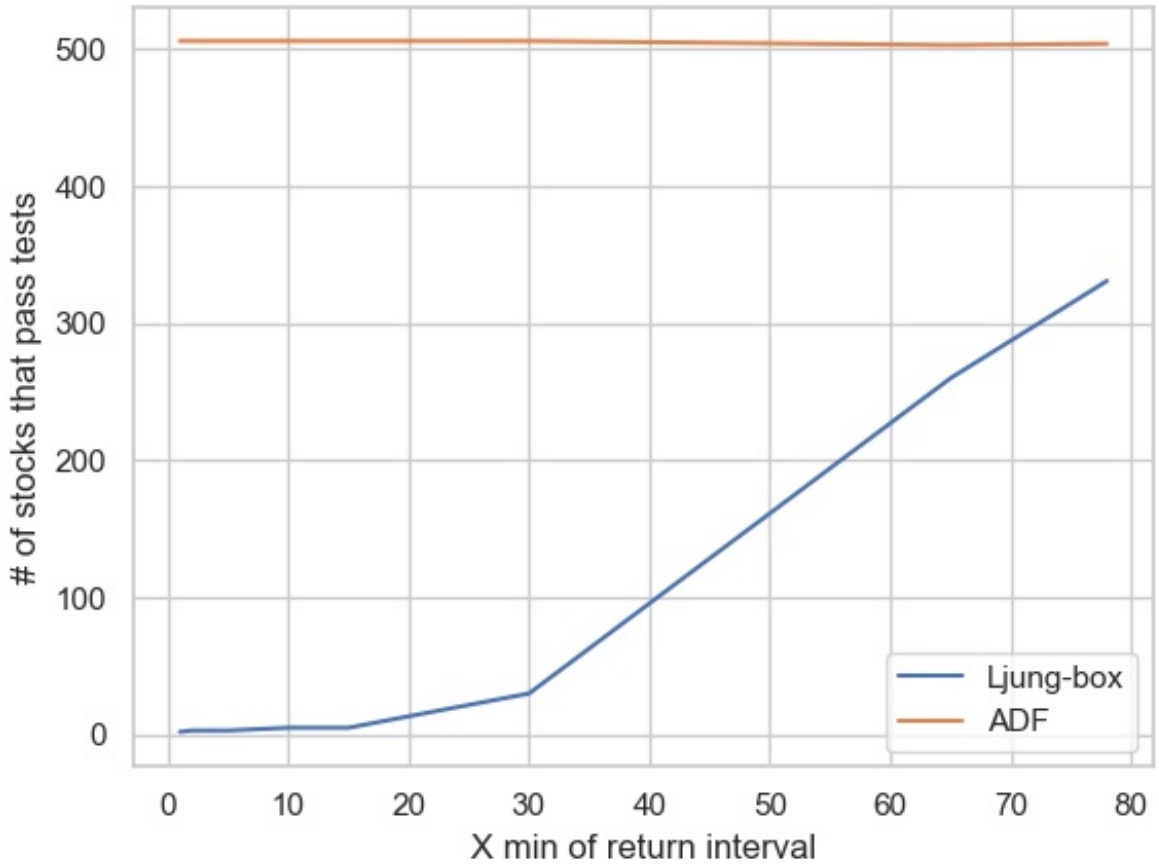
All figures are saved and can be found in `plot/stats`.

Analysis of Autocorrelation

We solve the frequency trade-off that should result in a timeframe size which is not too long and wasteful, while not too short to carry statistical behavior [1]. Specifically, we find a relative small frequency argument `xmin` at which the returns of a majority of stocks do not exhibit autocorrelation.

In detail, we use grid search on `xmin` and perform Ljung-box test on the returns computed from trades data for each stock, count the number of stocks whose null hypothesis is not rejected at a 5% level (i.e. cannot reject the autocorrelation-less hypothesis), and plot the counting results. We pick the number of lags from 1 to `390/xmin` or 5, whichever is smaller, to test if there is autocorrelation in terms of daily intervals, as well as `390/xmin` for seasonality, under the assumption that all outstanding orders will be liquidated before the closing hour each day.

As a bonus, we also perform augmented Dickey-Fuller test, which is to detect the existence of unit roots, and count the number of stocks that do not have a unit root. Both curves are plotted in a single graph below.



We observe that, even for `xmin=78`, only about 70% of stocks get rid of autocorrelation. We do notice that, in the literature `xmin` is usually chosen between 5 and 20 to prioritize the sample size over statistical effect. So, instead of pinning down a unique bucket size, we propose to dependent our answer to the specific application. For models that do not have "episodes" (i.e. end of trading day does not imply clearance), we accept `xmin=78` as our bucket size. For other models that do require daily settlement, we would rather pick `xmin=30`, to ensure we have enough observations while keeping the influence of autocorrelation low, since it acts as the reflection point in the graph.

Optimization

CVXOPT Example

This optimizer solves a quadratic programming problem in the form of:

$$f(x) = q^T x + \frac{1}{2} x^T Q x$$

Fin optimal x to minimize f(x), subject to restrictions:

$$
\begin{aligned}
Ax &= a \\
Bx &\leq b \\
x &\geq 0
\end{aligned}
$$

In particular, this optimizer in this problem solves the optimal portfolio weights minimizing

$$g(x) = \frac{1}{2}\mu(x^T \Sigma x) - p^T x$$

Subject to constraints:

$$
\begin{aligned}
Ax &= 1 \\
Gx &\leq 0
\end{aligned}
$$

μ is the risk aversion coefficient. The higher μ is, the more an investor hates risk. Risk is represented by the variance. We need to minimize the sum of variance and minus portfolio return. Σ is the corresponding covariance matrix of different kinds of equities.

p is the matrix made up of the expected return of different equities.

A is a matrix of shape (1, n) made up of 1s. G is a matrix of shape (n, n) with 1s in the diagonal line. The meanings of the restrictions is that no short is allowed and the sum of all the weights should be 1.

This example connects the solution of the optimal weighted when the risk aversion coefficient varies from 1 to 10000. We can easily see that when portfolio variance is 0, which means that the investor has a really high risk-aversion coefficient, all the weights are allocated to x4, which is the risk-free asset.

According to the description of the qp function written in function description, we can find that this optimizer has absolute tolerance of 1e-7 or absolute tolerance of 1e-6. The solution must satisfy at least one of these two criterions, or there is no solution.

CAPM analysis & Turnover

If we are in a CAPM world. The following assumptions are assumed to hold. All investors:

1. Aim to maximize economic utilities (Asset quantities are given and fixed).
2. Are rational and risk-averse.
3. Are broadly diversified across a range of investments.
4. Are price takers, i.e., they cannot influence prices.
5. Can lend and borrow unlimited amounts under the risk-free rate of interest.
6. Trade without transaction or taxation costs.
7. Deal with securities that are all highly divisible into small parcels (All assets are perfectly divisible and liquid).
8. Have homogeneous expectations.
9. Assume all information is available at the same time to all investors.

The market portfolio is represented by the S&P index and S&P index is a cap-weighted index. So, stock price and shares outstanding on June 20, 2007 and September 20, 2007 are needed. So, we calculate equity i's weight by:

$$W_{i,t} = \frac{P_{i,t} S_{i,t}}{\Sigma_i P_{i,t} S_{i,t}}$$

$P_{i,t}$ means the price of equity i at time t. $S_{i,t}$ means shares outstanding at time t.

We calculate the weight of individual stocks separately June 20, 2007 and September 20, 2007. To calculate turnover, we track the absolute difference of the weights between these two dates and divide it by 2 to get the turnover. We conclude that the turnover is about 4.44%. User can get the weights as well as the turnover by running `mvo/CAPM.py` we provide.

---

1. Also see **arXiv:2006.09611v2** ↩