Eve Online is a Massive Multiplayer Online Role Playing Game. Players are set in the far future and tasked with piloting spaceships to make money (ISK) and do whatever they desire. Players are able to come together to form corporations, then corporations of like minded players form alliances. Eve online has a completely player driver market, that means everything is created and destroyed within the game with very few external injections. The space in Eve online is divided into two main categories, High security space which is safe with an ingame police force and Null security space where outlaws roam free. Each of these groups are then further broken down into regions then constellations, then individual solar systems.

The data analyzed was all market data for one region in the month of January with the following attributes:

Type\_id: The specific code associated to each unique item

Region\_id: The specific code to region of space

Volume: The amount that a specific item was traded in one region

Order\_count: How many individual buy and sell requests were placed on the market combined

Highest: The highest price an item sold for

Lowest: The lowest price an item sold for

Average: The average price an item was sold for

Highlow\_difference: The difference between the highest and item sold for and the lowest

In the dataset there were 151,983 values each with the 9 attributes.

Key findings:

The total number of individual items traded was 57,737,048,074 across 2,392,903 orders making the average volume per order: 24,128 units.

The item that was traded the most was item 34:

Average high: 4.24

Peak high: 4.50

Average low: 3.98

Peak low: 3.78

Average average: 4.10

Average highlow difference: .26

Total Volume: 27,133,333,258

With this data we can make some observations:

Let’s say you bought 10% of the volume for resale. If you bought at the average low, and sold at the average high you would have:

Purchase price: 10,799,066,633

Sell price: 11,504,533,298

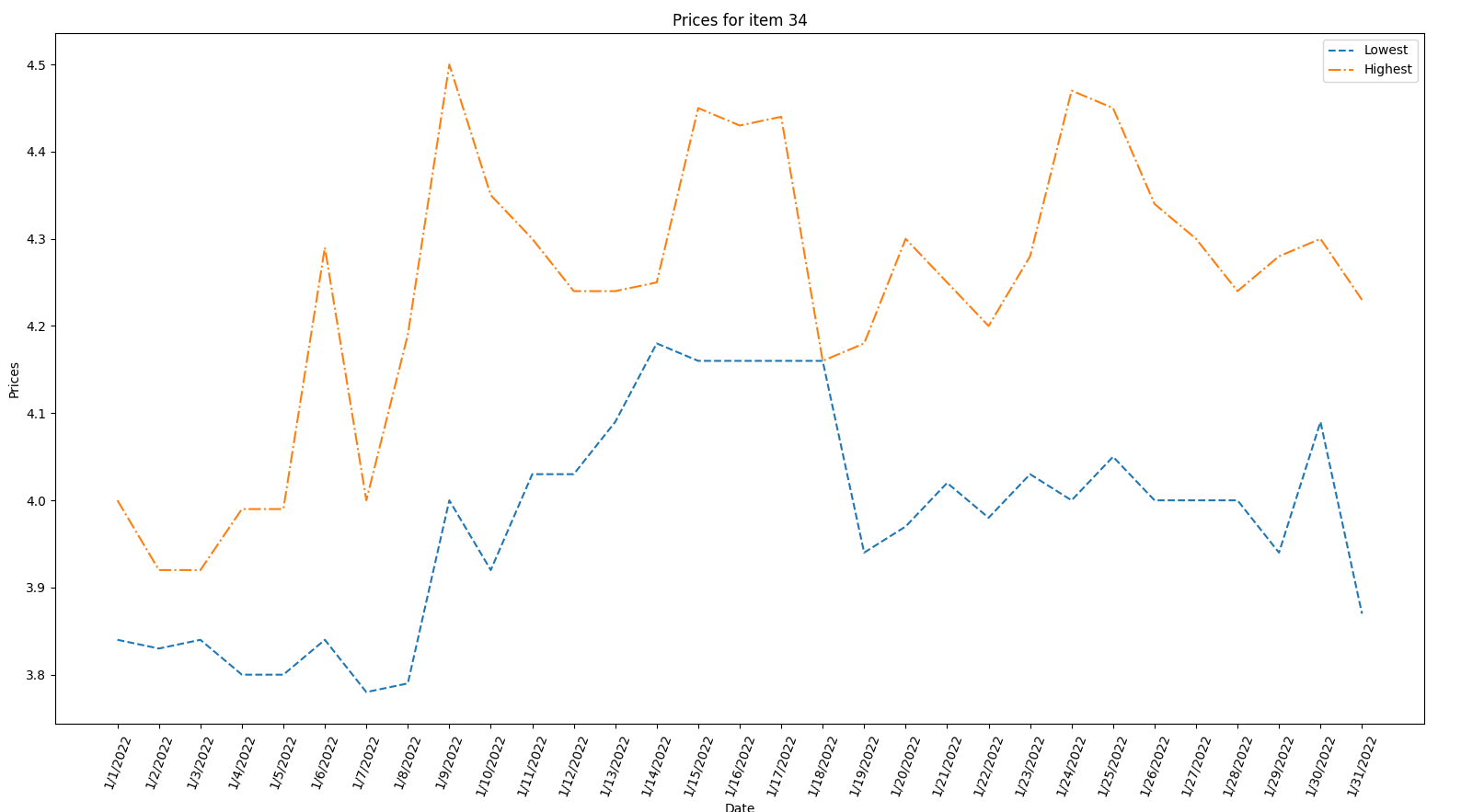
Profit: 7.2%

Now let’s say you watch trends and were able to do the same purchase at lowest price and sell at the highest:

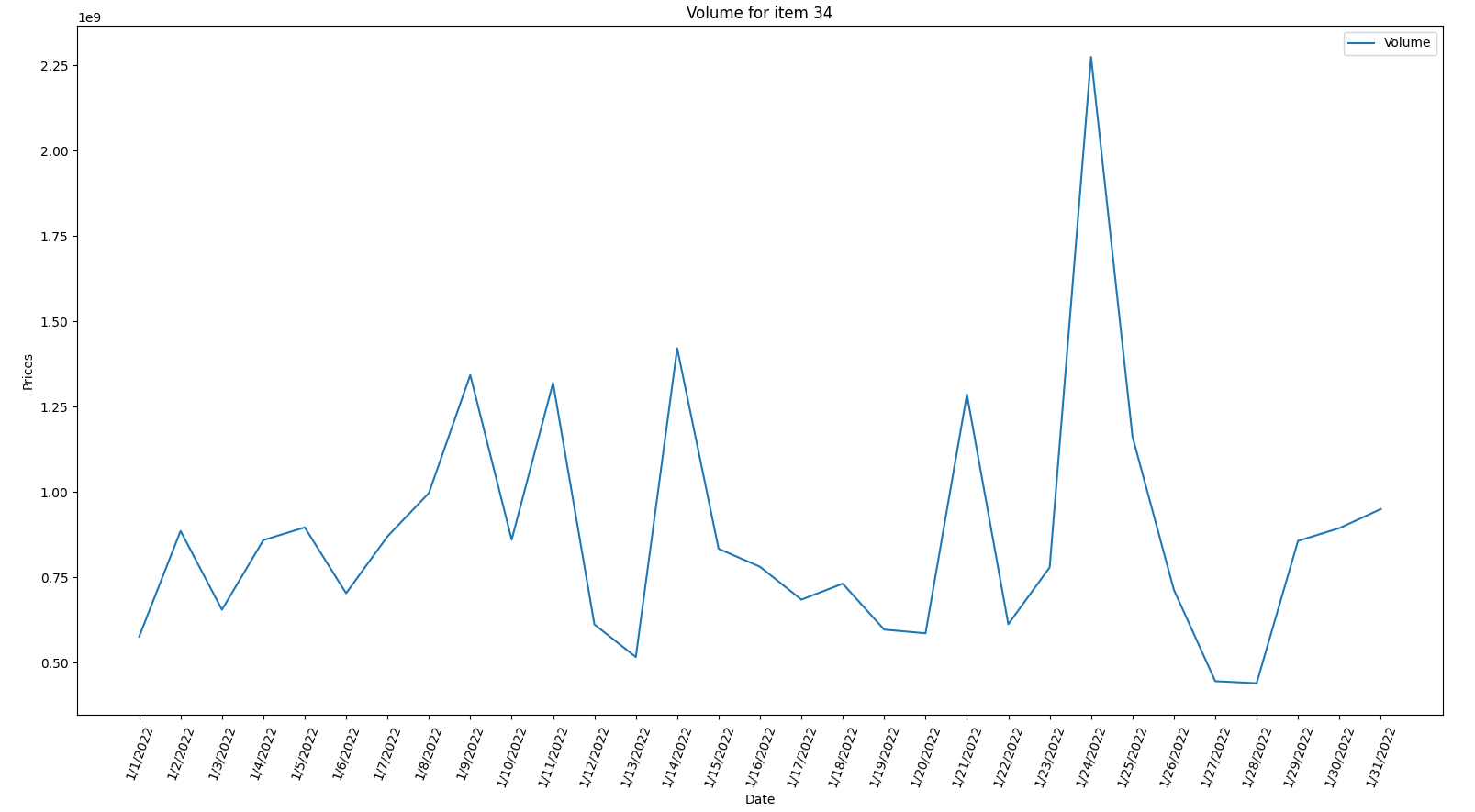
Purchase price: 10,256,399,968

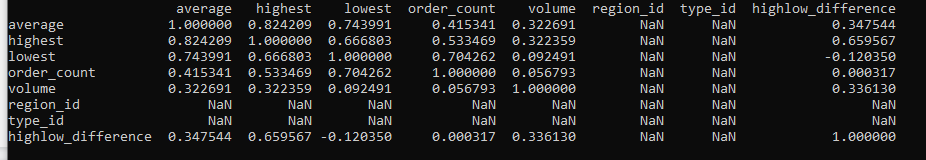
Sell price: 12,209,999,962

Profit: 16%



As you can see from the chart if you were to build this as you go with any single item you would be able to view and take advantage of trends.





So by way of visualization and statistics we are able to observe some other interesting facts.

The first being that contrary to standard economics a larger volume did not have a strong correlation between prices. There is a weak positive correlation between average price and volume indicating that when volume increases so did prices.

The second is the strong correlation between order\_count and lowest. This is telling us that the lowest prices were observed when the greatest number of unique orders were placed. We could use this to our advantage by driving prices down by placing multiple small orders rather than trying to make fewer large purchases.

Let's look at something with less fluctuation, item 28850 was the least traded item.

Average high: 15,426,250,000

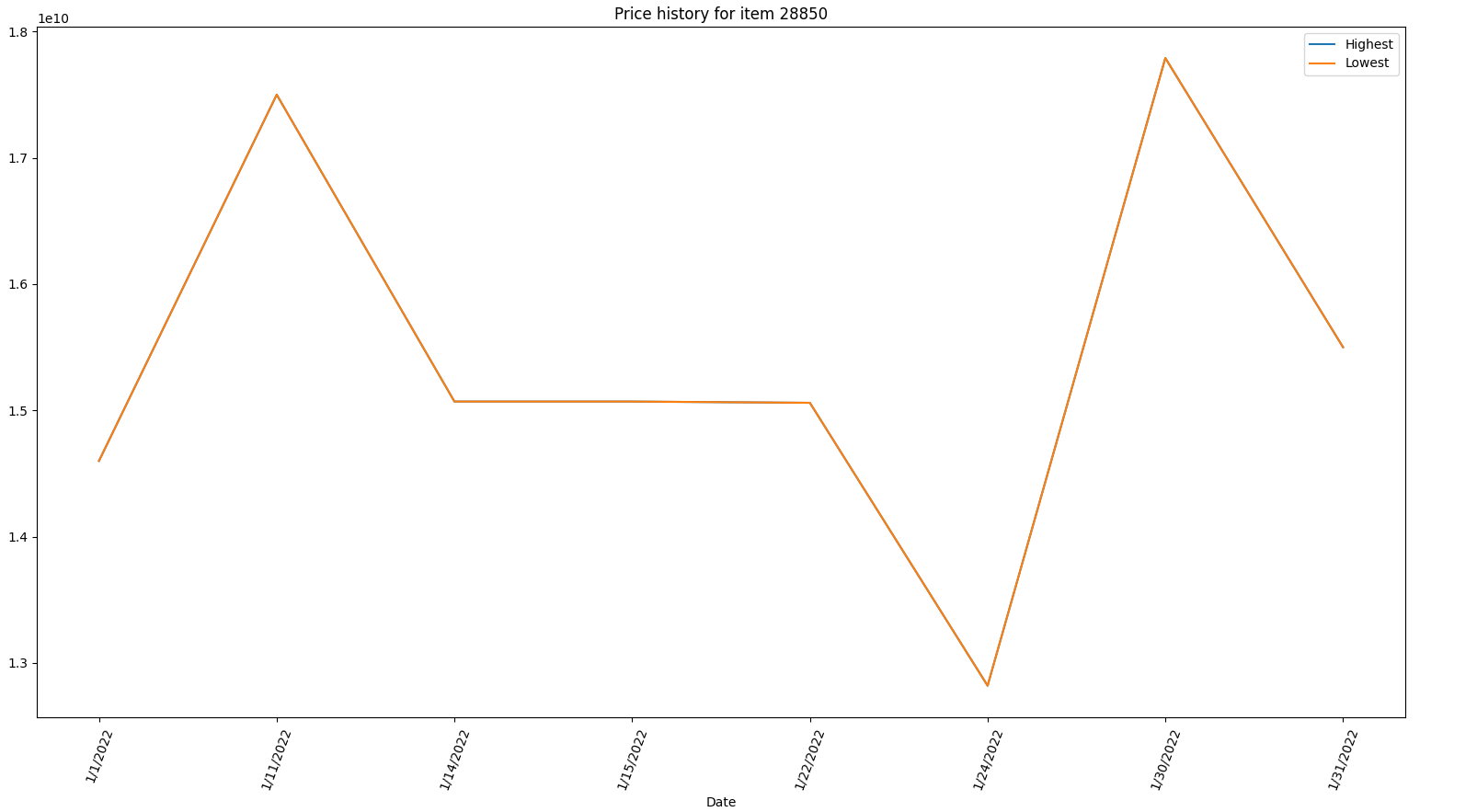
Peak high: 17,790,000,000

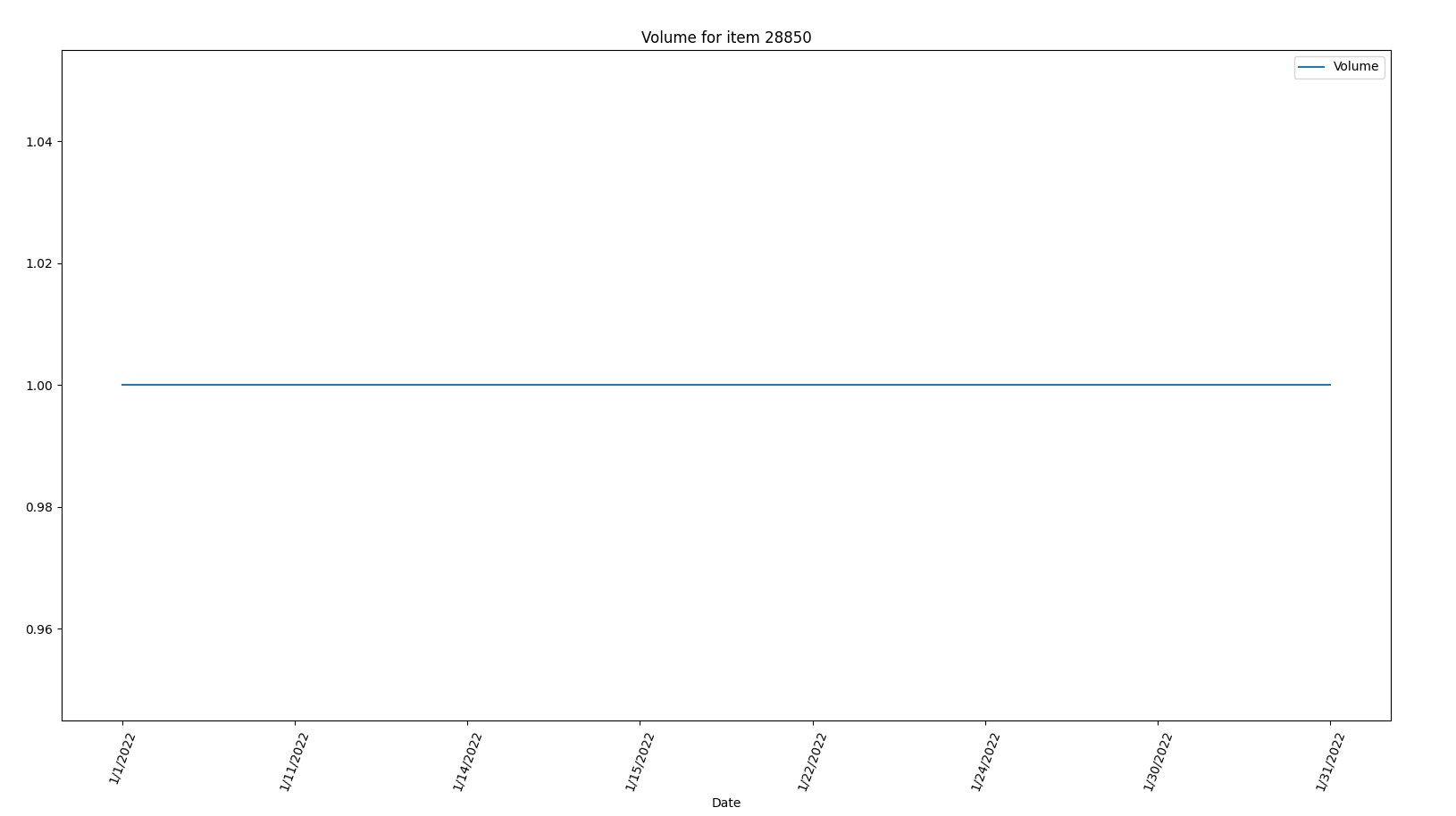
Average low: 15,426,250,000

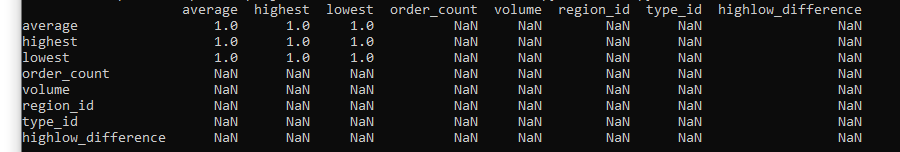
Peak low: 12,820,000,000

Average average: 15,426,250,000

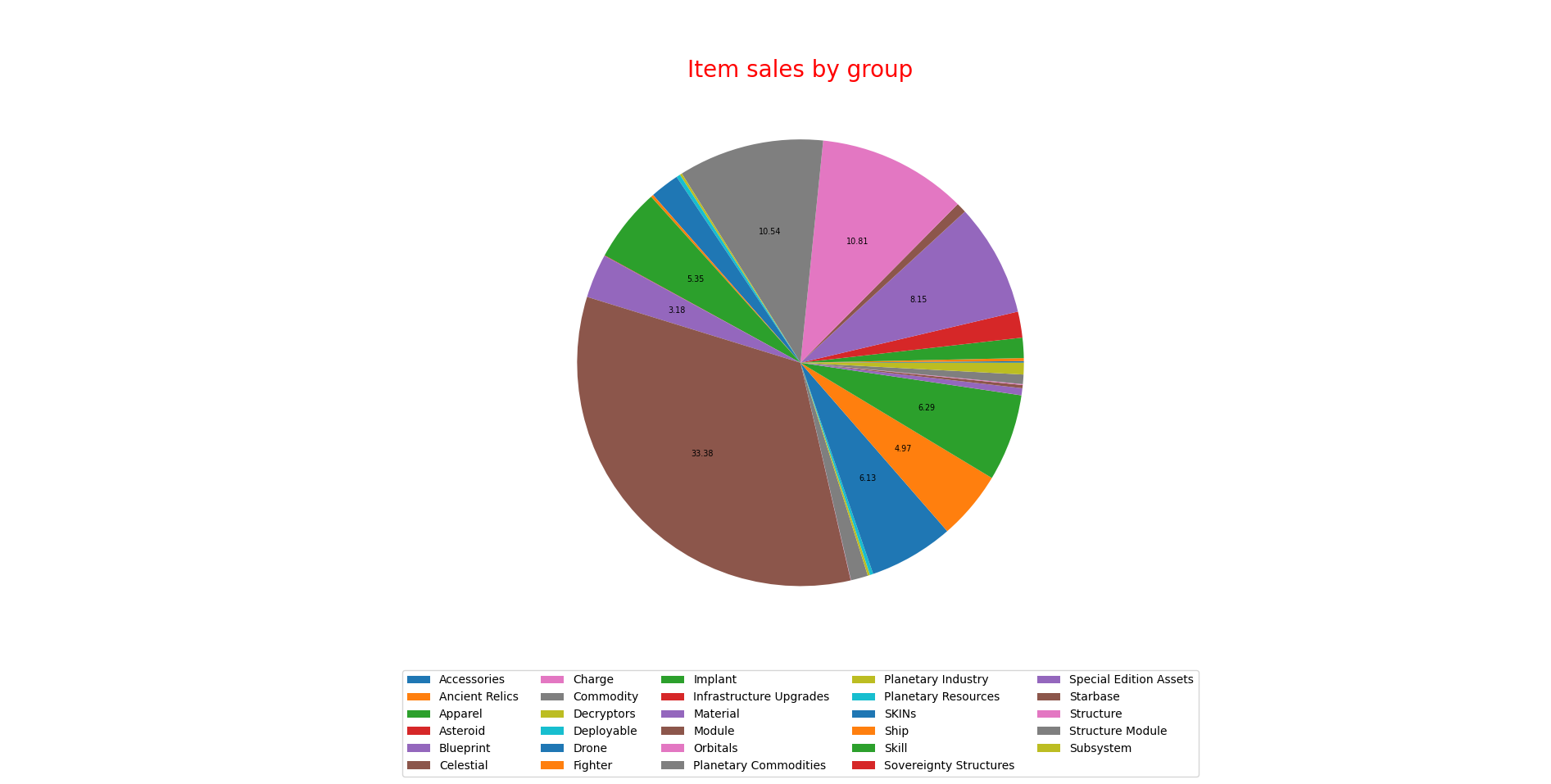
Average highlow difference: 0.0

Total Volume: 8

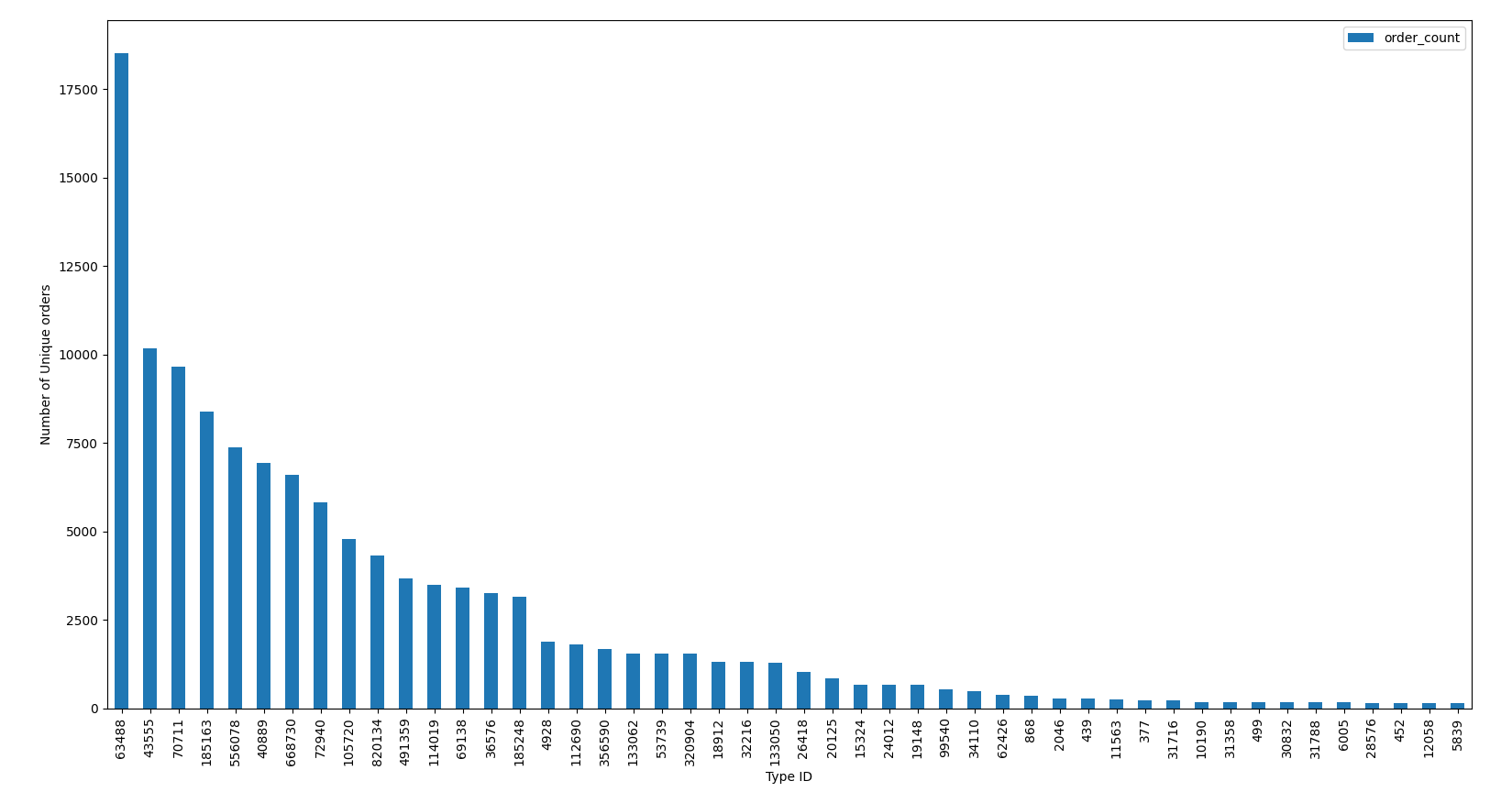


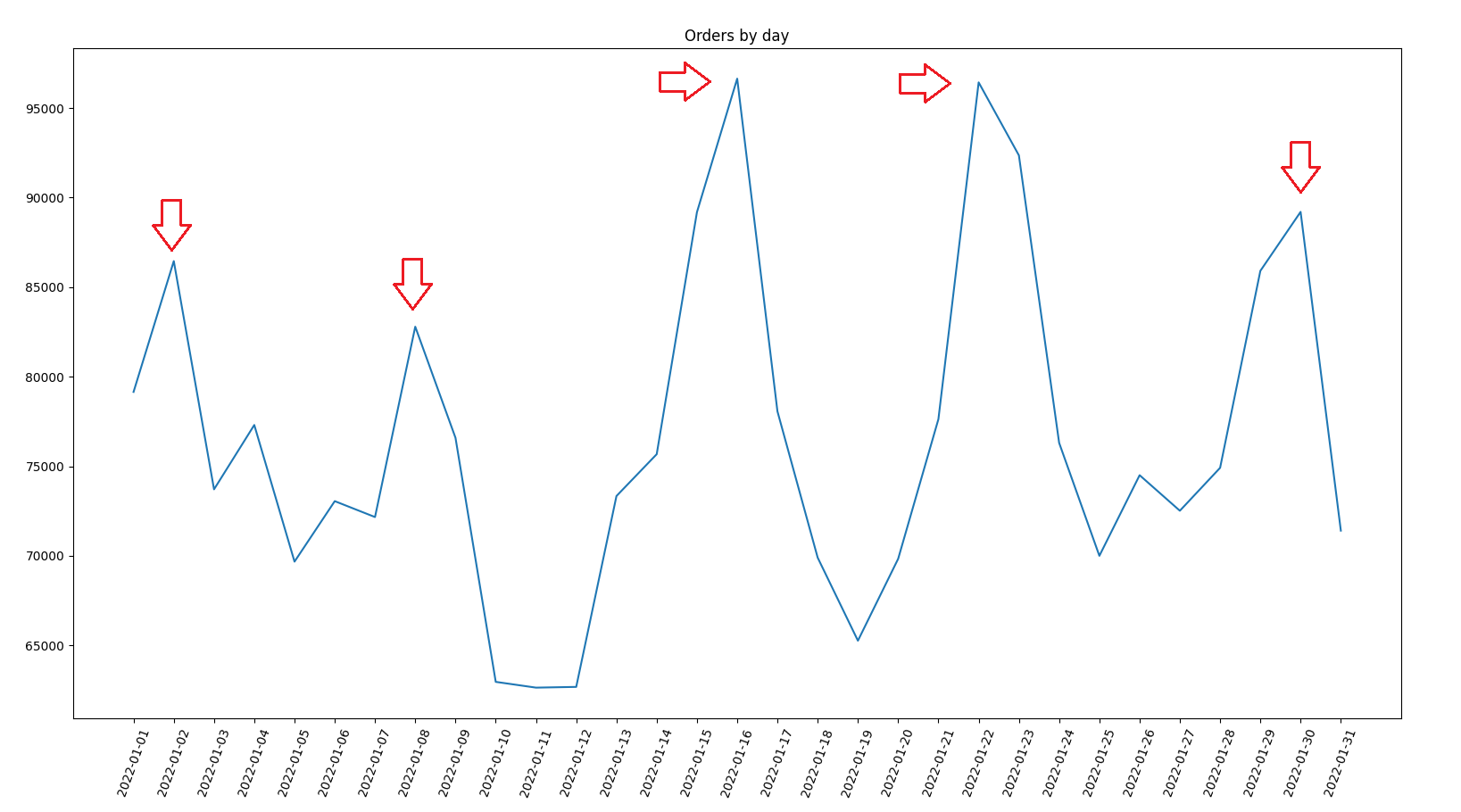
The difference between this item and the previous is that here we have no hourly fluctuation. As shown by the “highest” value and the “lowest” value being equal for the whole month. If we were to do the same as the previous example, and buy at peak low and sell at peak high our profit would be 4,970,000,000 or 28%.

With this item our correlation table isn't able to give us any useful information, one of the cons of having a small sample size.

Let's take a look at the distribution of categories.

From this we can see that our most popular item group was “Module” followed by “Commodity” and finally “Charge” totaling 54.73% of all sales.

Diving into the module category we are able to break down the top 1% of sales based on order\_count:

Of the top 1% of sales there were 122,121 unique sales across 48 different items. With this information we are able to see the best representation of a “high demand” item.

We are also able to figure out which days have the highest number of unique orders and determine that they are all weekends showing a correlation between day of the week and demand.

Now analyzing the data with some statistics I wanted to find out if Items with lower volume have a higher high/low difference. In order to determine “High” and “Low volume'' I picked the outer 15% of all volume. I decided to use a one tailed t-test for my statistics.

| High/Low Difference | Low volume | High Volume |
| --- | --- | --- |
| Number of observations | 7,335 | 7,335 |
| Mean | 6,259,450 | 1,223,207 |
| Standard Deviation | 32,189,369 | 8,554,7776 |
| Variance | 1,036,172,890,788,937 | 73,184,194,834,541 |

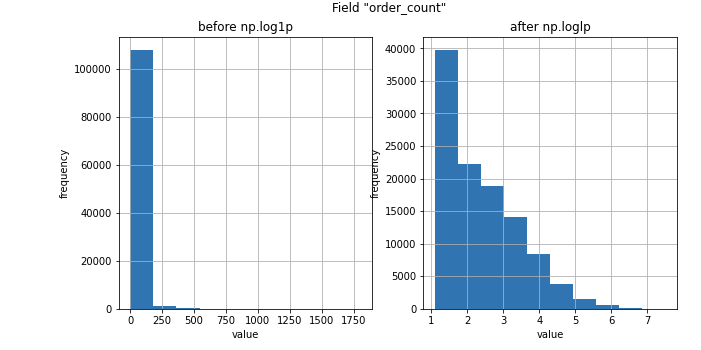
t= 12.949, df=14,668, p<.05

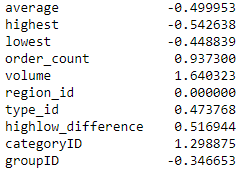
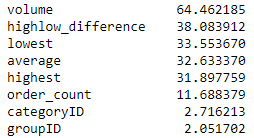
Since my t value had to exceed 1.96 we accept the research hypothesis as true and reject the null hypothesis.

Linear Regression results:

In our first unsupervised machine learning method we used Linear regression:

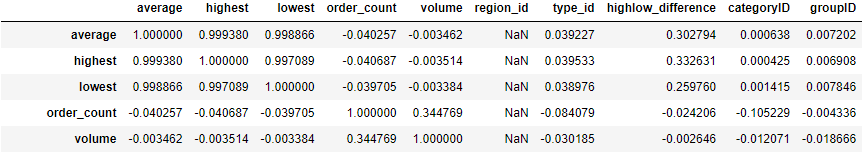
Initially we did no data cleaning and no feature encoding and our model was horrible. After some post analysis we observed that our data had no normal distribution which makes it significantly harder for an LR model to predict anything.

Our first step was applying a log-transformation to our data. 

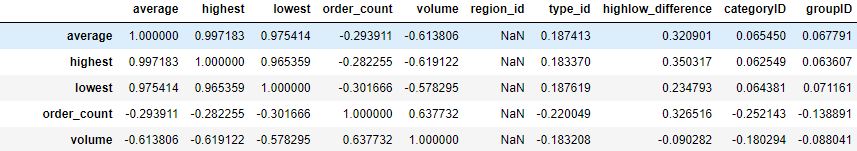
Here is our pre transformation skew values compared to our post.

As you can see we were able to bring the skewness of our variables to a much more reasonable level. This will both improve our model and allow us to more clearly see the correlation between two variables.

Here is the correlation table before the transformation:



After:



Our data is better but still needs some work to be ready for the model, here we introduce one-hot encoding. One hot encoding is a process by which categorical variables (names and words) are converted into a form that is better for a ML algorithm.

Here we applied the encoding and transformed our data from 151,599 rows and 14 columns for 2,122,386 unique values to 151,599 rows and 749 columns for 113,547,651 unique values.

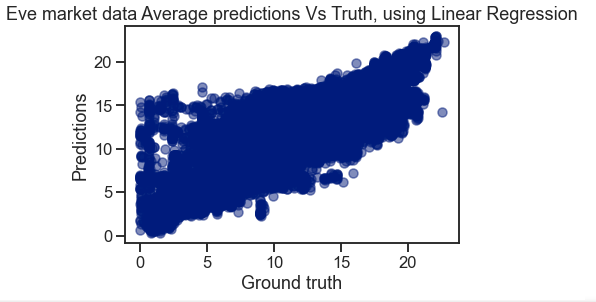
Finally our data is ready, we took a random sample of data for testing, trained our mode on it and ran it again on our hold out data. The following are test metrics to understand how well our model does.

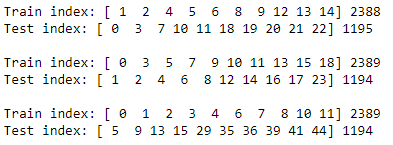
Mean Squared Error: 4.9221

With the standard being the closer the number is to zero the more accurate it is.

R2: 0.7176

With the standard for R2 the closer to 1 the greater the correlation.

Here is the results visualized:

One of the ways we can validate our models accuracy is to run multiple train/test splits. For the next model we ran three unique splits.

We know these to be accurate because even if the train indexes are the same every split has unique test values.