ASSIGNMENT TRIDGE PART 2

- What additional information, not asked in the previous exercise, do you think can be extracted from this dataset ?

Besides the trends, we also can extract additional information such as the average price from this dataset. By aggregating the price and average at different levels such as the country, region, grades or variety which could be very interesting for any trader to understand when the best timing is to invest on variety of a product and where to find it. The standard deviation can be also calculated using the mean value (the deviation of from the lowest and biggest value) to measure how widely prices are dispersed from the average price.

- How would you collect or extrapolate such information ? Please describe a vague algorithm/implementation you would use.

First, I will calculate the monthly average price for every row:

sum(weekly prices) / # of weekly prices per month).

Then I will aggregate it by grouping the average prices at the specified level and do the average again:

sum(group of monthly average prices) / # of element by group

- Assume this same dataset structure now contains 200,000,000 rows, would your logic scale ? If not, how would you make it scale? If yes, how does it handle the scale?

If the dataset structure is larger, we need to scale our logic to avoid an over-consummation of the machine (mainly memory and procs). One of the first solutions that will come up to save the memory is the chunking method (by assuming you do not need the entire dataset in memory all at one time). In our case, there are different ways to chunk:

- **Using Pandas library, the function read_csv() takes a chunk size parameter specifies the number of rows per chunk to process.**
- **Using multiprocessing library to set up chunks in a task queue**

For multiple CSV files processing, it would be more interesting to divide into different clusters (like Hadoop/Spark) which is a collection of computer to perform these kinds of parallel computations on big data sets.