Practical case – Reordering sunglasses for an online retailer

Case

- Retailer has been selling certain glasses model for 4 years
- Each order takes 3 months (average) to arrive
- Only 1 order may be sent every month

Current solution

The current procedure consists of an algorithm that calculates the average sales from the beginning of the dataset to the last entries, uses it to estimate the future sales keeping in mind the time restrictions (considering the time until arrival of the previous order -forecast leadtime- and the time until next arrival -forecast planning horizon-), and combines this estimation with the current stock and the expected product arrivals to decide the order size.

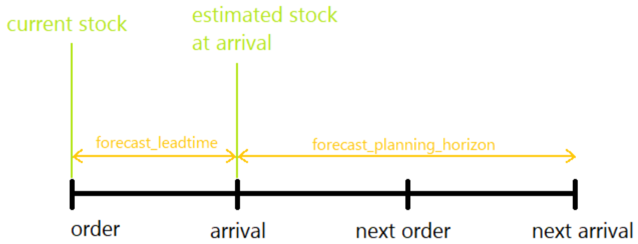


Figure 1

Problems

The ordering algorithm seems correct, being the forecasting function the most concerning one. The reason is that using the average is not optimal for this particular selling pattern. If we plot the data, we can see that the demand is seasonal, being the peaks of demand near June and December.

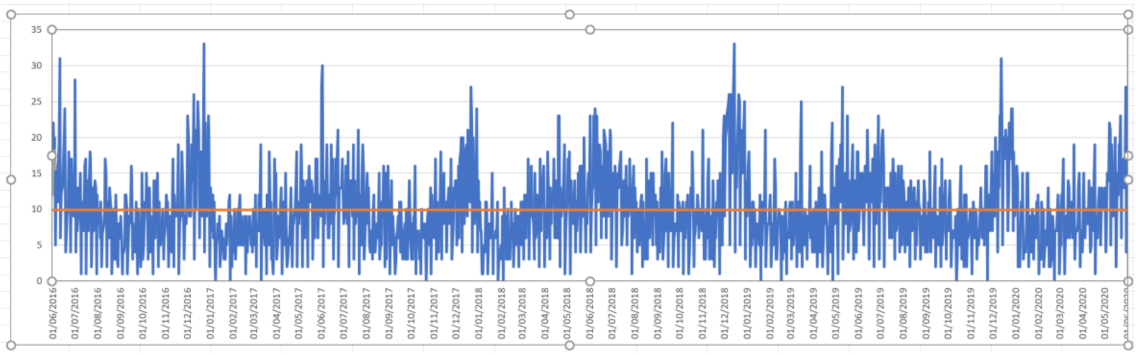
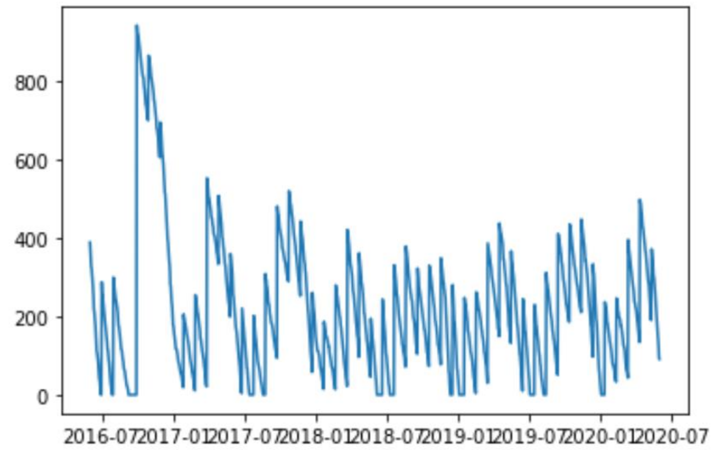


Figure 2

With a simple script, we can set up an experiment on how the stock would vary with this data and the current algorithm.




```
Losses due to lost sales: 16890.00€
Costs due to stock excess: 2689.96€
Total costs derived from stock management: 19579.96€
```

Figure 3

This algorithm is contained in the function test_algorithm of the code attached, and it applies the case's restrictions to keep track of the stockage. It also calls the function calc_losses, which prints out the losses due to the two main factors that are directly related to the stock control: losses due to lost sales because of shortage of products, and the costs of having products in the warehouse. This function is useful to have a metric to compare the different solutions.

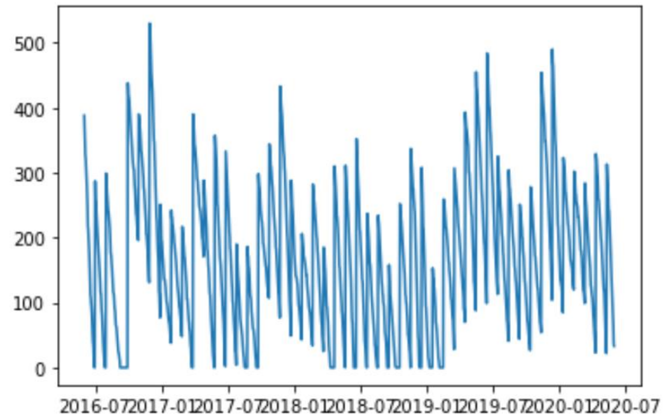
Some assumptions were made to develop these algorithms. The main ones are:

- Holding costs are linear; for instance, if you have one unit stored half a month, it costs 0.125€. If these are paid, for example, at the end of the month depending on how many units are left, it would need to be modified.
- Financial costs are supposed irrelevant for this case.
- There is no data of storage, order amounts or order dates before the example in the code with an order on June 1st, 2020. In order to proceed with the testing, the data provided for that date was used in the tests, assuming that on June 1st, 2016, the stock level was 400 units, the days until the next order were 30 and there were 600 units in transit, that would reach the warehouse in stacks of 300 the next 2 months consecutively.

The issues are very apparent if we look at *Figure 3*: because of the seasonality, there are overstockages certain months, and a lack of stock other ones. This derives from the usage of the total average in the forecasting algorithm.

Solution

There are several ways we could improve this behaviour. Since deliver time is a requirement for this situation, we developed an easy, straightforward algorithm to predict the demand and used it to substitute the old forecasting function. Since there is no tendency of the sales (if we look at *Figure 2*, the mean sales are constant throughout the whole period of time of which we have data), there is no need to make a complex regression model. The proposed solution takes advantage of the repeating demand from one year to the next one, and predicts that the demand on a period of time of a year will be the same as the next year in that same period. If we take a look at the results, we can see that there is a 2798€ savings (14.3%) -split in both the lost sales and the stock excess-, and the seasonality of the stockage almost disappears.




```
Losses due to lost sales: 14730.00€
Costs due to stock excess: 2051.14€
Total costs derived from stock management: 16781.14€
```

Figure 4

Improvements

This is a simple yet effective approach, but there is still room for improvement.

Even though we assume that there is no tendency of the sales, this is not true. Besides seasonality, there is a high chance that there is some general tendency (this could be calculated with the regression parameters), and it could be applied to the model.

The variance due to seasonality may not be the same as the year before. The model could include some indicators of tendency (if demand is growing or lowering at that moment of the forecast and by how much).

Most of the costs are derived from the lost sales because of stock shortage. It would help to add a security coefficient to the model to avoid these, since not selling a unit is more expensive than having an extra one. It also has hidden effects, such as fidelity loss of customers, impact on the brand image, wasted space on the warehouse…

Regarding the loss function, which we have defined as the total cost, there would be a mathematical minimum. Since the profit margin of a single unit always compensates for its storage cost of one month, the losses due to lost sales would need to be 0. Regarding to the costs of the stock in the warehouse, there would need to be as few units as possible, complying with the previous condition. This means that at every arrival, the stockage would need to be 0, but only at arrival, never before (that would mean lost sales). With these conditions, the minimum possible cost may be calculated.