McGill University

Desautels Faculty of Management

MGSC-670 – Revenue Management

Markdown Strategy

Presented to:

Prof. Maxime Cohen

By:

Samuel Dion, 261050552

Ali Al Samuraee 260739157

Uzair Ahmad 261008635

1 Introduction

In the retail business, revenue management includes budgeting, product pricing, and demand forecasting, as well as tactics for implementing markdowns and determining where initial prices should be dropped to improve sales over time. The first known strategy is promotional discount, which are usually short and targeted at certain client segment. Nonetheless, we are interested in a second strategy, which is called a markdown. **The latter occurs when a merchant permanently reduces the price of a product with the goal of maximizing revenue.** This technique is used as overbuying is a constant dilemma in the retail business, and markdowns solve this problem while also increasing income.

The retailer manager is responsible of selling 2,000 brand-new trendy wear products. Ultimately, **the objective is to determine the most profitable pricing approach**. The item's list price is $60, and it must be sold for at least one week at that price. Following that, the item may be lowered to $54 (a 10% discount), $48 (a 20% discount), or $36 (a 40% discount). The price cannot be raised after it has been reduced. The markdown season lasts 15 weeks, thus additional goods cannot be purchased during the sales season due to extensive production wait times. At the conclusion of the sales horizon, any things that remain will be disposed at a cost of $0.

To play this game, follow the link below: *http://www.randhawa.us/games/retailer/nyu.html*

The Retail Markdown Game is used in this research to determine the best strategy to handle markdowns by experimenting with different algorithms/approaches. An excel file was provided to us (Sales-Data.xlsx) and contains historical data to help us build a strategy.

2 Data Discovery

There are four columns in the historical sales data (Sales-Data.xlsx):

- Week: The first through the fifteenth week of the season (i.e. 14 decisions)
- Price: The amount charged in dollars on a certain week
- Sales: The quantity sold on that week
- Remaining Inventory: Inventory that is still available at the conclusion of the week.

In fact, through a series of visualization techniques, we were able to gain some interesting insights. Manifestly, as the price decreases, the number of sales increases, which gets in line with microeconomic concepts. We can also note that as the price decrease so does the dispersion of sales. For instance, on one hand, with a price of 36$, 50% of sales are between 137-238. On the other hand, at price of 60$, 50% of sales are 53-84. (*See Appendix: Histogram of different sales per price point*)

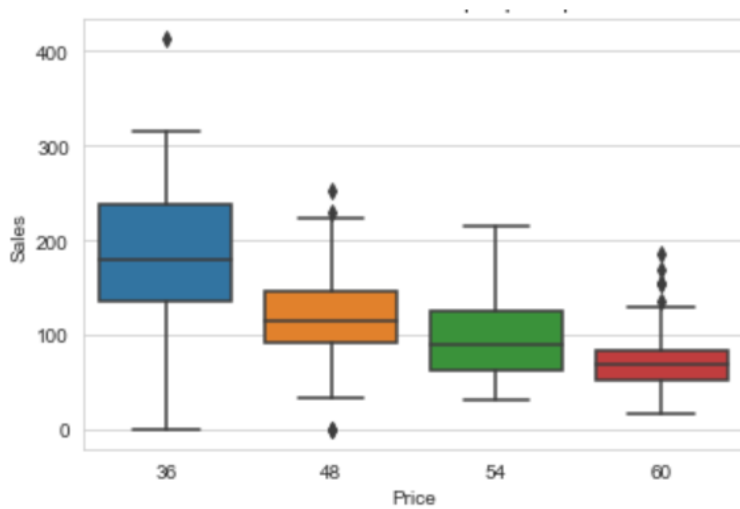


Figure 1: Number of sales in relation with selling price

We were also intrigued by the way that the number of sales behaved during time. We noticed that based on our scatterplot, the first diminution in price arrived in week 7, usually between weeks 8-10 (48$ and 54$). The earliest the price was adjusted to bottom 36$ was in week 10, with one-time sales being above 400 units, which seems like an outlier. Overall, we can say that our scatterplot showcases wise decisions, where there was no precipitation to sell low.

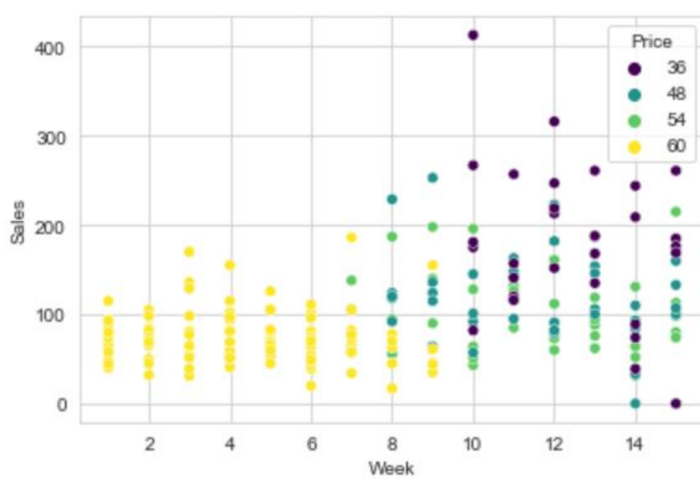


Figure 2: Number of sales in relation of the number of the week

3 Testing our strategies

To test our strategies, we have implemented a simulation process on the actual game by using Selenium. Inspiration for the implementation is derived from *this source*. Our final code and testing can be found in this GitHub *here*.

4 Baseline approaches

4.1 Maintain Price

To get a sense of how well (or poorly) our approaches perform, we wanted to use a *baseline model* for comparison purposes. The baseline approach is simple, as it maintains the price for all the weeks and no markdowns are performed. This gives an average difference vs an optimal foresight strategy of **21.7%** after playing 50 games. The standard deviation is 10.4% and we can expect an average difference between 18.8% and 24.7% within a 95% confidence interval.

4.2 Human playing the game

We were also interested in getting a feel of how well we would perform as human just looking at the price and applying markdowns on what we see. For that, we only performed 10 runs and we were able to get an average difference of **6.8%** with a standard deviation of 2.3%. This is obviously better than just maintaining the price constant. Results in this excel file (link)

5 Dynamic Approaches

5.1 Slope Estimates Ignorant of Previous Sales Data

Our first intuition was to do something really similar to what a human does when playing the game, meaning that we look at the general slope of the price and we take a decision whether we maintain the price or not solely based on if we are going to sell everything on week 15 and not before. This method doesn't consider the historical 15 runs data, but it does use data from our intuition when playing the game. This technique can be an exploratory approach to determine pricing for products where we do not have historical demands.

First, we maintain the price for 3 weeks to get an average estimate of the sales trend, using the historical sales information. We then project this slope to calculate if we are within 2 standard deviations of sales at the end of our 15-week mark. In case that our sales are low, we cut down the price by 10% & continue the same strategy to estimate the sales using the average sales (by maintaining the price for 2 runs and calculating slope). In case, we are not projected to meet our target, we then drop the price further by 20%. We continue this strategy until the 15th week-mark or we run out of inventory.

This approach yields average difference of **12.4%** with a standard deviation of 3.5% after playing 50 games. The 95% confidence interval is between 11.4% and 13.4%. We can see that this method is a hard coded version of following the human approach without prior knowledge of the data. It also performs worse than a human.

5.2 Linear Optimization with Regression Estimates

The linear optimization approach is theoretically sounder from a statistical/optimization perspective compared to the prior approach which didn't consider historical sales data. We are using the data from the previous 15 runs to build a linear regression model to predict the number of sales at a given week for a certain price (see results in Appendix). We are then incorporating the coefficients of this regression in a Linear Optimization Problem where we are trying to maximise the revenues at week 15 using the predicted sales at each week while respecting the game's constraints.

The formulation of the optimization can be found in the Appendix. The optimization is dynamic in the sense that for every week, the decision variable matrix changes size and the new LP is solved with the previously observed sales.

The results for this approach are the best with an average **6.4%** difference from the optimal solution. The standard deviation is equal to 2.8% and the 95% confidence interval is between 5.4% and 7.0%. However, we observed while running the model that it was very aggressive on doing early markdowns and was performing poorly on the "high sales" games which we think is due to the training data being from games with lower average sales.

6 Final Strategy and Future Improvements

From our experiment, only the Linear Optimization Problem provided better results than a human playing the game. Summary table of the results seen below:

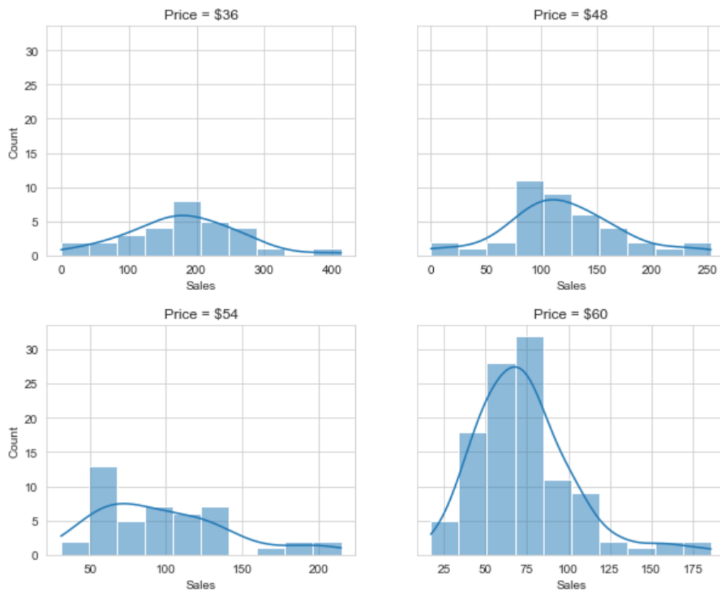
Approach	Mean	Standard Deviation	Confidence Interval	
			Lower bound	Upper bound
Maintain	21.7%	10.4%	18.8%	24.7%
Human approach	6.8%	2.3%	5.4%	8.2%
Slope Optimization	12.4%	3.5%	11.4%	13.4%
Linear Optimization	6.4%	2.8%	5.4%	7.0%

We believe that the problem with the other approach is that it has no prior information of the game and so doesn't react very well to sudden changes in slope. Also, this approach is based on our intuition of playing the game and so could not outperform us playing it. Although, manual tuning of this strategy could yield better results, this goes against the spirit of this revenue management exercise where, in real life, you don't have the chance to learn the "game" multiple times.

Thus, the LP Optimisation is the better approach but could be improved. For one, we think that its strength is when the sales distribution would be normally distributed at every week based on the price tag. However, what we are seeing with this game is that there seems to be a different price-response function for each game and our optimisation doesn't consider that. For example, if the sales are very high early on, the model doesn't adjust its forecast of sales for price i at week j, it still thinks that the sales at week $j + 1$ would follow the training data distribution. For future improvements, we should look for a way to either adjust our linear regression model prediction by retraining it with each runs data or have some sort of percent adjustment on the forecasted sales based on what we are seeing in the previous weeks.

7 Appendix:

Histogram of different sales per price point:



Linear regression results:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  Sales   R-squared:                       0.390
Model:                            OLS   Adj. R-squared:                  0.382
Method:                 Least Squares   F-statistic:                     47.16
Date:                Mon, 09 May 2022   Prob (F-statistic):           1.35e-23
Time:                        03:23:23   Log-Likelihood:                -1181.7
No. Observations:                 225   AIC:                             2371.
Df Residuals:                     221   BIC:                             2385.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    643.7429    112.455      5.724      0.000     422.122     865.363
Price         -9.4510      1.897     -4.983      0.000     -13.189      -5.713
Week         -23.8619      9.183     -2.599      0.010     -41.959      -5.765
Price:Week     0.3816      0.158      2.418      0.016       0.071       0.693
==============================================================================
Omnibus:                       27.942   Durbin-Watson:                   1.455
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               66.579
Skew:                           0.558   Prob(JB):                     3.49e-15
Kurtosis:                       5.420   Cond. No.                     1.63e+04
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.63e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Linear optimization formulation:

The formulation dynamically changes every run where the size of j will be adapted based on the remaining weeks to be completed and optimized.

Decision variable:

$$X_{ij} \rightarrow \text{binary variable - price i at week j}$$

Objective Function:

$$\max \sum_{j}^{weeks} \sum_{i}^{4} X_{ij} \times p_i \times \left(\beta_0 + \beta_1 * p_i + (\beta_2 + \beta_3 * p_i) \times week_j\right)$$

- $week_j$ is just the week # based on the number of times the game has been played. The linear regression expects a week number between 1 and 15.
- p_i is the price for row i so one of [60, 54, 48, 36]
- $\beta_{0,..,4}$ are the regression coefficients from the linear regression model.

For every week, we sum the revenues for each i where the only non-negative price will be the price selected and we repeat for the number of weeks.

Constraints:

1 price needs to be selected every week:

$$\sum_{i}^{4} X_{ij} = 1 \qquad \forall j \text{ in weeks}$$

Maximum of 2000 sales:

$$sales = \sum_{j}^{weeks} \sum_{i}^{4} X_{ij}\left(\beta_0 + \beta_1 * p_i + (\beta_2 + \beta_3 * p_i) \times week_j\right)$$

$$2000 - \sum PastSales - \text{sales} \geq 0$$

The sum of past sales are the sales completed during the run of the game.

Can't go back to a higher price once a markdown selected.

$$X_{i-1,j-1} \leq \sum_{k}^{i-1} X_{k,j} \quad \forall i \in [1 \text{ to } 4] \quad \forall j \in [1 \text{ to weeks}]$$

weeks are the remaining weeks left based on the number of weeks already completed in the game.