MKT382 – Team Project 1

Optimal Retail Pricing

Store-level Scanner Data on Orange Juice Sales

In this project you will apply regression to estimate the effect of prices and promotions on unit sales of different orange juices using store-level scanner data. You will use the results to make inferences about pricing and promotional decisions.

Dominick's Finer Foods, a major supermarket chain in the Chicago area owned by Safeway, has expressed interest in micro marketing strategies. They intend to customize prices to local market conditions. Your team has been contacted for a pilot project to see if and how Dominick's pricing strategy can be improved upon. The focus of this project is on prices of one product category, i.e., orange juices, in one store. Your team will write a report making recommendations on Dominick's pricing strategies for orange juices.

Orange juice is a relatively stable category with moderate new product introductions and heavy promotion activity. It is a large volume category, accounting for 1% of total store sales. This category is more highly developed in urban areas, despite lower income levels, which may be due to convenience and fewer kids in such areas. For example, New York City has a 400 index for orange juice (the average market has an index of 100).

Your assignment

Your task is to analyze store-level scanner data, taking note of the instructions below, and to write a report. Your report will cover around 2-3 pages of text and refer to tables and graphs that you include in an appendix. Please use the following paragraph headings for your report (taking into account the guidelines 1 through 4 below): 1) Introduction, 2) Method, 3) Limitations, 4) Results, 5) Conclusions, 6) Appendices.

The data

Store-level scanner data from the Orange Juice Category has been made available to you. The database was collected from the weekly store scanner data and contains information about unit sales, price, feature, display, and gross margins for each item in the category. These items were aggregated up from an individual level of Universal Product Code (UPC).

The data you are being given is for a single store. The most important items in the category are:

1. Tropicana Premium 64 oz
2. Tropicana Premium 96 oz
3. Tropicana 64 oz
4. Minute Maid 64 oz
5. Dominick's 64 oz private label

These are the most popular Stock Keeping Units (SKUs) in the category. The data are stored on the course website in a tab-separated text file called Dominicks_scanner_data-orange_juice.txt (Download the file by clicking the link and save it to your desktop).

The header (first line) of the text file contains the variable names for each column of the data and have the following prefixes:

WEEK	Week number (First week is indicated by week 40, which is 06/14 - 06/20)
SALES	Sales in ounces
PRICE	Price (Standardized to $ per ounce)
DISPLAY	In-store display bonus-tags (Indicator variable, corrected for volume)
FEAT	Was the item featured in a weekly brochure? (Indicator variable, corrected for volume)
GRMAR	Gross margin

The suffix for each of these variables denotes the measure for that particular SKU (i.e., PRICE2 is the price for the second SKU in the category, i.e., Tropicana Premium 96 oz., etc.).

Gross margins are given in proportions and vary depending on the prices.

Guidelines to the analyses

1. Descriptive statistics

You are to load the dataset in R and compute descriptive statistics (such as sample mean and variance) and plots using the scanner data. From these descriptive statistics (i.e., average sales) compute the market shares of each SKU (in unit and dollar sales) for **all** of the SKU's and briefly describe the market. What are the leading products? Compare the pricing and promotional strategy for each of the products. Do you expect all the brands to be competing in an identical way? Pay extra attention to visualizing your results.

Hint

You can use the `read.table()` function to load the data from the text file as a `data.frame`:

```
data = read.table(file='Dominicks_scanner_data-orange_juice.txt', header=TRUE, sep='\t')
```

2. Regression

Linear regression can be used to build a series of demand models that predicts sales as a function of promotional variables. You are to build such a model. As discussed in class, demand models relate sales to price and promotional variables. You want to consider competition, cannibalization, post-promotion dip.

Hint

- Make sure to transform sales and prices to a log scale, so the corresponding coefficients reflect elasticities. So you will run five regressions, taking as a dependent variable the log sales of each SKU.

- There is likely to be competition and cannibalization in this market. Specify cross-price elasticities in your model as well.
- Think about modeling and testing for post-promotion dips for price cuts.

In your report describe the following

- Interpret the coefficients in the regression equation (conceptually, in words, not the numbers)
- Discuss model fit
- Discuss the value of the coefficients among the different brands. *Refer to the class notes on demand models for details.* More specifically,

  - ✓ Discuss competitive effects and cannibalization. Which SKU is the most sensitive and which SKU is the least sensitive to price changes (i.e., has the largest and smallest slope terms for its own price)?
  - ✓ Discuss the implications of cannibalization for manufacturers and the retailer separately. Their incentives are different.
  - ✓ Which product is the most sensitive to features and in-store displays?
  - ✓ What do the parameter estimates tell you about the relationship between the sales of a product and the price of other products (i.e., how do price changes of a product influence sales of another product)?
  - ✓ Do you see demand exhibit a post-promotion dip? Why?
  - ✓ Do all significant parameters have the expected signs?

3. What-if analysis

Estimation of the regression models helps to obtain general insights. Furthermore, a retailer can use the estimates of the regression coefficients to predict product sales for different price levels and promotions. Using cost information, we can even predict the profitability of different strategies. Use the regression coefficients to predict sales and profitability for the following price and promotion strategies.

Instructions

- Store the regression coefficients calculated under part 2 and create R code that predicts the sales as a function of prices and promotions (basically, each prediction is a linear combination of the predictor variables you specified in your model and the corresponding regression coefficients). Then compute the expected category profits as a function of predicted sales and costs. Note that the gross margins depend on the prices charged, so for this analysis, use the average cost per unit for each SKU as an estimator for the actual cost. You can compute the costs using the following relationship:

$$grmar = (p - cost)/p$$

- Evaluate the following pricing/promotional strategies and predict sales, market shares and profits (assume costs of displays and features are negligible):

  - ✓ The current pricing strategy by using average historical prices of the brands; no displays or features.
  - ✓ What is the incremental sales and profits when Dominick’s private label is on display and Tropicana Regular 64oz on feature?
  - ✓ Tropicana is interested to run a display for their regular 64oz item and lower price to $1.60. They are offering a discount which results in a gross margin of .27. What is the expected

profitability (holding all other prices constant, i.e., set to average values and assuming that no other feature or display is held).

✓ Try two more scenarios of your choice and discuss the results.

4. Profit maximization

Now we will determine the prices that optimize category profit for the retailer, building upon the simulator you built under 3. In order to do so, we can use a constrained optimization package called `Rsolnp` to optimize prices (or any other optimization package of your choice). We will take the following into account:

- Fix the overall average price in order not to hurt the price image of the store (this will stabilize the solutions as well).
- We will also set the prices such that we are not predicting outside of the data interval (i.e., the optimal prices cannot be lower or higher than those we have observed in the scanner data). Keep displays, features, and post-promotion effects out of the consideration).

Instructions on optimization

- You can use the `solnp` function from the `Rsolnp` package for constrained optimization
- Install Rsolnp package: `> install.packages("Rsolnp")`
- Load the library: `> library("Rsolnp")`
- You have to specify the objective function `fun`, and a function the equality constraint and two constraints for upper and lower bounds. The following shows the arguments for `solnp()`:

```
results = solnp(prices0,  # initial values for parameters
                fun=cal_profit,  # objective function
                eqfun=const_eqn,  # equality constraint function
                eqB=c(0),  # value equality function should equal to
                LB=prices_min,  # lower bounds of parameters
                UB=prices_max,  # upper bounds of parameters
                coefs=coefs,  # additional arguments
                control=control)  # options for optimizer
```

- Objective function (`fun`)
    - Given data and parameters, this function should calculate and return the profit. Typically optimizers minimize the objective function by default, so return negative profit.
- Equality constraint
    - For equality constraint, you have to specify the `eqfun` argument, the function which evaluates the equality constraint (it should return the difference between the mean of the prices and mean of the historical prices), and `eqB`, the values which each constraint equals to. That is, if your equality constraint is `p1 + p2 = 3`, then `eqfun` returns the value of `p1 + p2`, and `eqB = c(3)`. (Somewhat confusing, as `eqfun` could just return the value of `p1 + p2 - 3` without `eqB`)
    - Your `fun` and `eqfun` must accept the same arguments. That means, if you use any additional arguments for `fun`, `...` (typically you would want to pass the data to the objective function as well), `eqfun` also must accept ..., even if it does not use it.
- Lower and upper bounds of parameters.
    - You need to find the minimum and maximum of the prices for each SKU. Unfortunately, R does not have `colMins` or `colMaxs` function. To get the minimum value for each column, use:

```
do.call(pmin, lapply(1:nrow(mat), function(i) mat[i,]))
```

For the maximum values:

```
do.call(pmax, lapply(1:nrow(mat), function(i) mat[i,]))
```

Report and discuss the optimal prices. How did the prices, sales, market shares and profits change? Do they have face validity? What are some of the main limitations of this approach? Discuss conceptually how you could change your methodology to overcome these limitations.