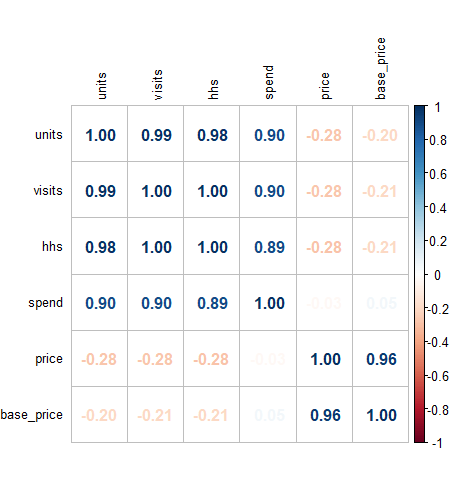
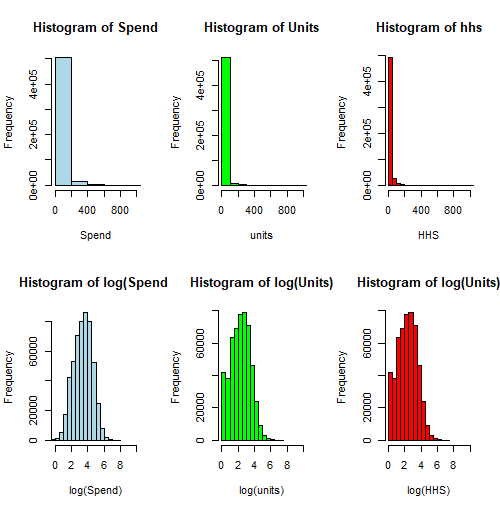
Data:

The data is at two levels: Level 1: Products (UPC, DESCRIPTION, MANUFACTURER, CATEGORY, PRODUCT\_SIZE), and Level 2: Stores (STORE\_ID, STORE\_NAME, CITY, STATE, SIZE, AVG\_WEEKLY\_BASKET). Hence, we must use multi-level analysis for this analysis.

**Feature Engineering:**

1. Merged data from products, stores, and transactions using upc, store\_id and store\_num.
2. Removed the oral hygiene segment as stated.
3. Converted upc, store\_num, city, state and segment into factors.
4. Removed rows with N/A.
5. Split the week\_end\_date to get the month and year.

Looking at the distributions of the 3 DV’s (Spend, Units and hhs)



The log transformed DV’s look approximately normal. We can use LMER models with log transformation on these DV’s since the data is of multi-level nature.

High correlation between price and base\_price variable and among the DV’s. Let’s drop base\_price variable to overcome multicollinearity.

Predictor Table-

|  |  |  |
| --- | --- | --- |
| **Predictor** | **Effect** | **Rationale** |
| *DV: Spend, Units, HHS* | | |
| Feature, Display, tpr\_only | + | Various types of promotional activities are expected to positively impact the sales performance of a store. |
| Segment | Unknown | It may be necessary to consider the distinct sales trends that could exist among value, mainstream, and upscale stores. Also required for Q2. |
| Category | Unknown | Different products might have different sales. Also required for Q2. |
| Avg\_Basket | + | More baskets sold might contribute to higher sales. |
| Store\_num | Unknown | This variable accounts for the store-specific, fixed, or random effects that contribute to the variability in sales across different stores. |
| State | Unknown | Sales among the 4 states could vary. |
| Excluded |  |  |
| product\_size | Unknown | could be correlated with upc as it can capture the size. Similarly, visits is correlated with all 3 DVs. |
| Base\_price | +/- | Base price due to correlation with price. |
| Manufacturer, Sub\_Category etc. | N/A | They are just descriptions or their effect is caught by the above included variables. |

Lmer Models-

lmer\_spend <- lmer(log(spend) ~ (feature + display + tpr\_only) \* (segment + category) + avg\_weekly\_baskets + price +

month + year + (1|store\_num) + (1 | state),

data = merged, REML=FALSE)

lmer\_units<- lmer(log(units) ~ (feature + display + tpr\_only) \* (segment + category) + avg\_weekly\_baskets + price +

month + year + (1|store\_num) + (1 | state),

data = merged, REML=FALSE)

lmer\_hhs<- lmer(log(hhs) ~ (feature + display + tpr\_only) \* (segment + category) + avg\_weekly\_baskets + price

+ month + (1|store\_num) + (1 | state),

data = merged, REML=FALSE)

The interaction term was introduced to answer Q2.

Output-

> stargazer(lmer\_spend,lmer\_units, lmer\_hhs, type="text", single.row=TRUE)

============================================================================================

Dependent variable:

--------------------------------------------------------------

log(spend) log(units) log(hhs)

(1) (2) (3)

--------------------------------------------------------------------------------------------

feature 0.296\*\*\* (0.015) 0.370\*\*\* (0.015) 0.381\*\*\* (0.015)

display 0.795\*\*\* (0.008) 0.871\*\*\* (0.008) 0.878\*\*\* (0.007)

tpr\_only -0.096\*\*\* (0.007) -0.021\*\*\* (0.007) -0.126\*\*\* (0.007)

segmentUPSCALE 0.035\*\*\* (0.011) 0.036\*\*\* (0.011) 0.035\*\*\* (0.011)

segmentVALUE -0.328\*\*\* (0.059) -0.325\*\*\* (0.059) -0.367\*\*\* (0.060)

categoryCOLD CEREAL 1.019\*\*\* (0.003) 0.940\*\*\* (0.004) 0.925\*\*\* (0.003)

categoryFROZEN PIZZA 0.443\*\*\* (0.006) 0.497\*\*\* (0.006) 0.371\*\*\* (0.006)

avg\_weekly\_baskets 0.00002\*\*\* (0.00000) 0.00002\*\*\* (0.00000) 0.00002\*\*\* (0.00000)

price -0.012\*\*\* (0.001) -0.303\*\*\* (0.001) -0.271\*\*\* (0.001)

monthApr -0.069\*\*\* (0.006) -0.071\*\*\* (0.006) -0.049\*\*\* (0.006)

monthAug -0.080\*\*\* (0.006) -0.082\*\*\* (0.006) -0.061\*\*\* (0.006)

monthDec -0.007 (0.006) -0.007 (0.006) 0.006 (0.006)

monthFeb -0.011\* (0.006) -0.014\*\* (0.006) 0.007 (0.006)

monthJul -0.108\*\*\* (0.006) -0.116\*\*\* (0.006) -0.089\*\*\* (0.006)

monthJun -0.078\*\*\* (0.006) -0.085\*\*\* (0.006) -0.064\*\*\* (0.006)

monthMar -0.123\*\*\* (0.006) -0.134\*\*\* (0.006) -0.115\*\*\* (0.006)

monthMay -0.066\*\*\* (0.006) -0.070\*\*\* (0.006) -0.051\*\*\* (0.006)

monthNov -0.060\*\*\* (0.006) -0.067\*\*\* (0.006) -0.054\*\*\* (0.006)

monthOct -0.051\*\*\* (0.006) -0.063\*\*\* (0.006) -0.044\*\*\* (0.006)

monthSep -0.079\*\*\* (0.006) -0.085\*\*\* (0.006) -0.061\*\*\* (0.006)

year2010 -0.030\*\*\* (0.003) -0.032\*\*\* (0.003)

year2011 -0.054\*\*\* (0.003) -0.056\*\*\* (0.003)

year2012 -0.233\*\*\* (0.016) -0.239\*\*\* (0.016)

feature:segmentUPSCALE -0.188\*\*\* (0.012) -0.183\*\*\* (0.012) -0.186\*\*\* (0.012)

feature:segmentVALUE 0.076\*\*\* (0.012) 0.073\*\*\* (0.012) 0.101\*\*\* (0.011)

feature:categoryCOLD CEREAL 0.296\*\*\* (0.016) 0.261\*\*\* (0.016) 0.181\*\*\* (0.016)

feature:categoryFROZEN PIZZA 0.442\*\*\* (0.016) 0.286\*\*\* (0.016) 0.274\*\*\* (0.016)

display:segmentUPSCALE -0.057\*\*\* (0.011) -0.067\*\*\* (0.011) -0.062\*\*\* (0.011)

display:segmentVALUE 0.025\*\* (0.011) 0.024\*\* (0.011) 0.046\*\*\* (0.011)

display:categoryCOLD CEREAL -0.202\*\*\* (0.011) -0.254\*\*\* (0.011) -0.271\*\*\* (0.010)

display:categoryFROZEN PIZZA -0.132\*\*\* (0.010) -0.255\*\*\* (0.010) -0.264\*\*\* (0.010)

tpr\_only:segmentUPSCALE -0.042\*\*\* (0.010) -0.043\*\*\* (0.010) -0.046\*\*\* (0.010)

tpr\_only:segmentVALUE -0.146\*\*\* (0.009) -0.145\*\*\* (0.010) -0.099\*\*\* (0.009)

tpr\_only:categoryCOLD CEREAL 0.144\*\*\* (0.009) 0.135\*\*\* (0.009) 0.228\*\*\* (0.008)

tpr\_only:categoryFROZEN PIZZA 0.324\*\*\* (0.010) 0.145\*\*\* (0.011) 0.262\*\*\* (0.010)

Constant 2.676\*\*\* (0.114) 2.568\*\*\* (0.112) 2.316\*\*\* (0.107)

--------------------------------------------------------------------------------------------

Observations 418,554 418,554 418,554

Log Likelihood -493,533.300 -498,160.100 -484,801.300

Akaike Inf. Crit. 987,144.600 996,398.100 969,674.500

Bayesian Inf. Crit. 987,571.400 996,825.000 970,068.500

============================================================================================

Note: \*p<0.1; \*\*p<0.05; \*\*\*p<0.01

Testing Assumptions:

The Linear Mixed Effects Regression (LMER) model is robust to Normality and Equality of Variance Assumption. We still must test multicollinearity and Independence.

Multicollinearity- Passed

|  |  |  |
| --- | --- | --- |
| > vif(lmer\_spend)  GVIF Df GVIF^(1/(2\*Df))  feature 12.517111 1 3.537953  display 3.940451 1 1.985057  tpr\_only 3.376714 1 1.837584  segment 1.066544 2 1.016236  category 4.786189 2 1.479100  avg\_weekly\_baskets 1.039516 1 1.019566  price 3.109802 1 1.763463  month 1.132733 11 1.005681  year 1.111811 3 1.017822  feature:segment 2.954547 2 1.311060  feature:category 20.634749 2 2.131326  display:segment 2.841690 2 1.298357  display:category 6.129187 2 1.573442  tpr\_only:segment 1.866758 2 1.168886  tpr\_only:category 3.008537 2 1.317009 | > vif(lmer\_units)  GVIF Df GVIF^(1/(2\*Df))  feature 12.517111 1 3.537953  display 3.940450 1 1.985057  tpr\_only 3.376714 1 1.837584  segment 1.066558 2 1.016240  category 4.786190 2 1.479100  avg\_weekly\_baskets 1.039487 1 1.019552  price 3.109802 1 1.763463  month 1.132733 11 1.005681  year 1.111811 3 1.017822  feature:segment 2.954577 2 1.311064  feature:category 20.634746 2 2.131326  display:segment 2.841720 2 1.298361  display:category 6.129185 2 1.573442  tpr\_only:segment 1.866782 2 1.168889  tpr\_only:category 3.008537 2 1.317009 | > vif(lmer\_hhs)  GVIF Df GVIF^(1/(2\*Df))  feature 12.504445 1 3.536163  display 3.940011 1 1.984946  tpr\_only 3.372308 1 1.836384  segment 1.066150 2 1.016142  category 4.775599 2 1.478281  avg\_weekly\_baskets 1.039238 1 1.019430  price 3.098627 1 1.760292  month 1.036856 11 1.001646  feature:segment 2.954401 2 1.311044  feature:category 20.524676 2 2.128478  display:segment 2.841493 2 1.298335  display:category 6.125656 2 1.573215  tpr\_only:segment 1.866676 2 1.168873  tpr\_only:category 2.999965 2 1.316070 |

What are the effects of product display, being featured on in-store circular, and temporary price reduction on product sales (spend), unit sales, and number of household purchasers?

* “Display” has the highest impact on sales compared to hhs and units increasing the "spend" at mainstream stores by 79.5%, UNITS by 87.1 %, and HHS by 87.8%.
* The “feature” results in a substantial increase in SPEND (29.6%), UNITS (37%), and HHS (38.1%) for bag snacks at mainstream stores.
* The “tpr\_only” results in a decrease in "spend" by 9.6%, UNITS by 2.1%, and HHS by 12.6% for bag snacks at mainstream stores.

How do the effects of display, feature, and TPR on SPEND vary by product categories (cold cereals, frozen pizza, bag snacks) and store segments (mainstream, upscale, value)?

d(log(spend))/d(feature) = 0.296 - 0.188\* upscale + 0.076\* value + 0.296 \* cold\_cereal + 0.442\* frozen\_pizza

d(log(spend))/d(display) = 0.871 - 0.057\* upscale + 0.025\* value - 0.202\* cold\_cereal - 0.132\* frozen\_pizza

d(log(spend))/d(tpr\_only) = -0.096 - 0.042\* upscale - 0.146\* value + 0.144\* cold\_cereal + 0.324\* frozen\_pizza

|  |  |  |  |
| --- | --- | --- | --- |
|  | Bag\_snacks | Cold\_cereal | Frozen\_pizza |
| Feature | 29.6 % | 59.2 % | 73.8 % |
| Display | 87.1 % | 66.9 % | 73.9 % |
| Tpr\_only | - 9.6 % | 4.8 % | 22.8 % |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Value | Mainstream | Upscale |
| Feature | 37.2 % | 29.6 % | 10.8 % |
| Display | 89.6 % | 87.1 % | 81.4 % |
| Tpr\_only | -24.2 % | -9.6 % | -13.8 % |

What are the five most price elastic and five least price elastic products? Price elasticity is the change in units sold for change in product price.

Top 5 most price elastic products:

> print(top\_5\_most\_price\_elastic)

# A tibble: 5 × 2

upc price\_elasticity

*<chr>* *<dbl>*

1 1111009507 1.82

2 1111009477 1.73

3 1111009497 1.18

4 1111085319 0.192

5 1111085345 -0.435

> cat("\nTop 5 least price elastic products:\n")

Top 5 least price elastic products:

> print(top\_5\_least\_price\_elastic)

# A tibble: 5 × 2

upc price\_elasticity

*<chr>* *<dbl>*

1 3000006610 -3.72

2 7218063052 -3.82

3 7218063979 -4.02

4 7192100336 -4.04

5 3800039118 -4.38

As the retailer, which products would you lower the price to maximize (a) product sales and (b) unit sales, and why?

In order to maximize SPEND, we should focus on products with the most negative price elasticity values. This means that a slight decrease in price for these products would result in a significant increase in both spend and units sold.

As a retailer, based on the output provided for the top 5 most and least price elastic products, the following strategies can be implemented to maximize product sales and unit sales:

(a) To maximize product sales:

Lower the price of the product with UPC '1111009507' which has a price elasticity value of 1.82.

Lower the price of the product with UPC '1111009477' which has a price elasticity value of 1.73.

Lower the price of the product with UPC '1111009497' which has a price elasticity value of 1.18.

These products are the most price elastic among the top 5, meaning that a slight decrease in price would result in a significant increase in product sales. By lowering the price of these products, it is likely to attract more customers and drive higher sales, thereby maximizing product sales.

(b) To maximize unit sales:

Lower the price of the product with UPC '3000006610' which has a price elasticity value of -3.72.

Lower the price of the product with UPC '7218063052' which has a price elasticity value of -3.82.

Lower the price of the product with UPC '7218063979' which has a price elasticity value of -4.02.

Lower the price of the product with UPC '7192100336' which has a price elasticity value of -4.04.

Lower the price of the product with UPC '3800039118' which has a price elasticity value of -4.38.