Statistical Data Mining

SNACKCHAIN FINAL PROJECT

Jenna Ptak, Shawn Murphy, Pablo Zumba

Table of Contents

Introduction

Our objective in the final project is to present our findings from analyzing sales and promotions data from a large retail chain. The sales data is comprised of over 500,000 transactions from 79 stores and 58 products belonging to four product categories (bagged snacks, cold cereal, frozen pizza, and oral hygiene products) over a period of 156 weeks. For this assignment, we have excluded products belonging to the oral hygiene category. In this paper we will be discussing the effects of the pricing and promotion strategies on total spending for that product, the number of households who purchased that product, the number of store visits, and the most elastic and inelastic products. At the end of the analysis, we are interested in answering the following questions:

1) What is the effect of promotions, displays, or being featured in the circular on product sales (spend), unit sales, and the number of household purchasers?
2) How do the above effects vary by product categories (cold cereals, frozen pizza, bag snacks) and store segments (mainstream, upscale, value)?
3) What are the five most prices elastic and five least price-elastic products? Price elasticity is the change in sales for a unit change in product price.
4) As the retailer, which products would you lower the price to maximize (a) product sales and (b) unit sales, and why?

Details of the data files are listed below.

File	Description	Columns
Stores	Contains information about each store location	Store ID, Store Name, City, State, Metropolitan Statistical Area, Store Type, Number of Parking Space,

		Square Footage, Average Weekly Baskets
Products	Contains information about each product that is sold	Universal Product Code, Description, Manufacturer, Category, Subcategory, Product Size
Transactions	Contains information about each transaction that is performed at each store	Week-End Date, Store Num, Universal Product Code, Units Sold, Number of Unique Purchases That Included The Product, Number Of Purchasing Households, Total Spend, Price, Base Price, If Product Was In In-Store Circular, If Product Was A Part Of In-Store Promotional Display, Temporary Price Reduction Only

Data Pre-Processing and Data Cleaning

We used the R Code below to clean and pre-process the data

```
rm(list=ls())
library(rio)

#Import SnackChain.Xlsx file.
stores_df       <- import("~/Downloads/SnackChain.xlsx", sheet="stores")
#79 stores
products_df     <- import("~/Downloads/SnackChain.xlsx", sheet="products")
#58 products
transactions_df <- import("~/Downloads/SnackChain.xlsx",
sheet="transactions") #524950 transactions
#Joining Tables.
temp <- merge(transactions_df, products_df, by.x=c("UPC"), by.y=c("UPC"))
#join by UPCs between transactions and products
df   <- merge(temp, stores_df, by.x=c("STORE_NUM"), by.y=c("STORE_ID"))
#join by adding the stores.
#Removing temporary Files.
rm(transactions_df)
rm(stores_df)
rm(temp)
rm(products_df)

#Data Preprocessing

#Remove oral hygiene products
df <- df[df$CATEGORY != "ORAL HYGIENE PRODUCTS", ]  #this could be for
another project

#Check for missing values
colSums(is.na(df))
```

```
#' 282548 missing PARKING values; we will not use this data since it is
higher (store) level,
#' 10 NA values for price and 173 for BASE_PRICE
df <- subset(df, select = -c(PARKING))
df <- df[complete.cases(df), ] #Removing missing values
View(df)

#Renaming columns to make them easier to understand
colnames(df)[colnames(df) == "STORE_NUM"] = "store_id"
colnames(df)[colnames(df) == "UPC"] = "product_id"
colnames(df)[colnames(df) == "WEEK_END_DATE"] = "week_ending_date"
colnames(df)[colnames(df) == "UNITS"] = "units_sold"
colnames(df)[colnames(df) == "VISITS"] = "unique_purchases"
colnames(df)[colnames(df) == "HHS"] = "purchasing_households"
colnames(df)[colnames(df) == "SPEND"] = "total_spend"
colnames(df)[colnames(df) == "PRICE"] = "price_product_charged"
colnames(df)[colnames(df) == "BASE_PRICE"] = "base_price_product"
colnames(df)[colnames(df) == "FEATURE"] = "circular"
colnames(df)[colnames(df) == "DISPLAY"] = "promotional_display"
colnames(df)[colnames(df) == "TPR_ONLY"] = "temp_price_reduction"
colnames(df)[colnames(df) == "DESCRIPTION"] = "product"
colnames(df)[colnames(df) == "MANUFACTURER"] = "manufacturer"
colnames(df)[colnames(df) == "CATEGORY"] = "category"
colnames(df)[colnames(df) == "SUB_CATEGORY"] = "subcategory"
colnames(df)[colnames(df) == "PRODUCT_SIZE"] = "size"
colnames(df)[colnames(df) == "STORE_NAME"] = "store_name"
colnames(df)[colnames(df) == "CITY"] = "city"
colnames(df)[colnames(df) == "STATE"] = "state"
colnames(df)[colnames(df) == "MSA"] = "metropilitan_area"
colnames(df)[colnames(df) == "SEGMENT"] = "type_of_store"
colnames(df)[colnames(df) == "SIZE"] = "store_sqft"
colnames(df)[colnames(df) == "AVG_WEEKLY_BASKETS"] =
"avg_weekly_baskets_sold"

#Converting Categorical variables into factor variables.
df$store_id = factor(df$store_id)
df$product_id = factor(df$product_id)
df$category = factor(df$category)
df$subcategory = factor(df$subcategory)
df$category = relevel(df$category, "BAG SNACKS")
df$city = factor(df$city)
df$state = factor(df$state)
df$type_of_store = factor(df$type_of_store)
df$type_of_store = relevel(df$type_of_store, "VALUE")
df$product = factor(df$product)


#Extracting year, month, and week_ending_date from date.
df$year = format(df$week_ending_date, "%Y")
df$year = factor(df$year)
df$month = format(df$week_ending_date, "%b")
df$month = factor(df$month)
df$month = relevel(df$month, "Jan")
```

```
df$week_num = difftime(df$week_ending_date, df$week_ending_date[1],
units="weeks")
df$week_num = df$week_num + 1
attach(df)
str(df)
View(df)
```

Data after cleaning:

	store_id	product_id	week_ending_date	units_sold	unique_purchases	purchasing_households	total_spend	price_product_charged	base_price_product	circular
1	367	1111009477	2009-01-14	13	13	13	18.07	1.39	1.57	0
2	367	7027316204	2009-06-03	15	9	9	26.25	1.75	2.47	0
4	367	1111085319	2009-01-14	14	13	13	26.32	1.88	1.88	0
5	367	7192100336	2010-02-24	1	1	1	6.99	6.99	6.99	0
6	367	1111085350	2009-01-14	35	27	25	69.30	1.98	1.98	0

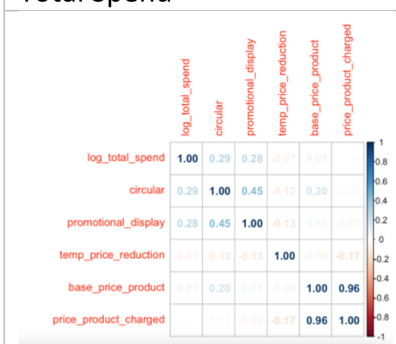
	promotional_display	temp_price_reduction	product	manufacturer	category	subcategory	size	store_name	city	state	metropilitan_area
0	0	1	PL MINI TWIST PRETZELS	PRIVATE LABEL	BAG SNACKS	PRETZELS	15 OZ	15TH & MADISON	COVINGTON	KY	17140
0	0	1	SHURGD MINI PRETZELS	SHULTZ	BAG SNACKS	PRETZELS	16 OZ	15TH & MADISON	COVINGTON	KY	17140
0	0	0	PL HONEY NUT TOASTD OATS	PRIVATE LABEL	COLD CEREAL	ALL FAMILY CEREAL	12.25 OZ	15TH & MADISON	COVINGTON	KY	17140
0	0	0	DIGIORNO THREE MEAT	TOMBSTONE	FROZEN PIZZA	PIZZA/PREMIUM	29.8 OZ	15TH & MADISON	COVINGTON	KY	17140
0	0	0	PL BT SZ FRSTD SHRD WHT	PRIVATE LABEL	COLD CEREAL	ALL FAMILY CEREAL	18 OZ	15TH & MADISON	COVINGTON	KY	17140

type_of_store	store_sqft	avg_weekly_baskets_sold	year	month	week_num
VALUE	24721	12706.53	2009	Jan	1 weeks
VALUE	24721	12706.53	2009	Jun	21 weeks
VALUE	24721	12706.53	2009	Jan	1 weeks
VALUE	24721	12706.53	2010	Feb	59 weeks
VALUE	24721	12706.53	2009	Jan	1 weeks

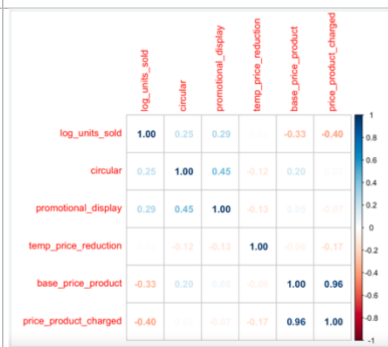
Data Visualization

Correlation matrix

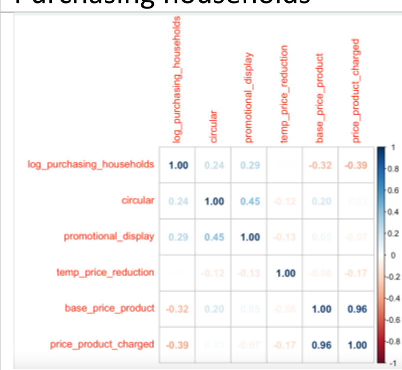
Total Spend



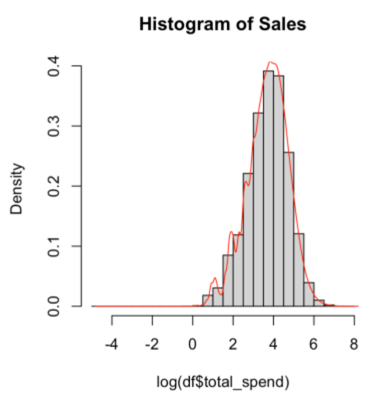
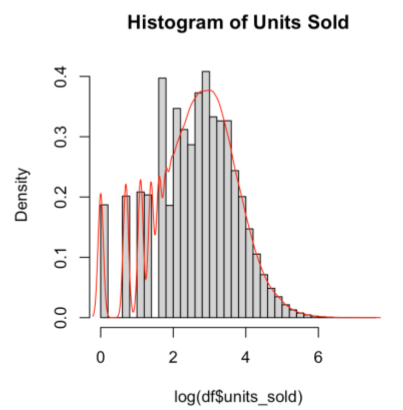
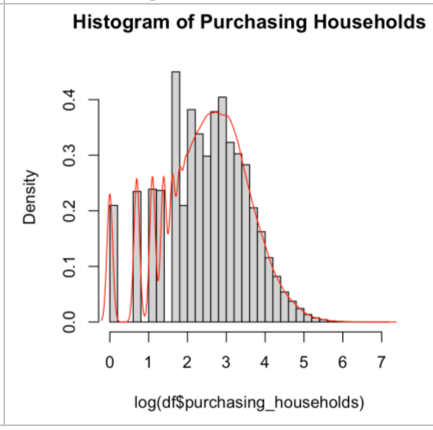
Total Units



Purchasing households



Histograms

Total Spend	Total Units	Purchasing households
		

Effect of promotional displays or being featured in the circular

PLM Models

Variable	Model	R2 Value
Total Spend	spendrandom <- plm(log(total_spend) ~ circular + promotional_display + temp_price_reduction + price_product_charged + week_num + category + month + type_of_store, data=df, index=c("store_id"),model= "random")	0.126
Total Units	unitsrandom <- plm(log(units_sold) ~ circular + promotional_display + temp_price_reduction + price_product_charged + week_num + category + month + type_of_store, data=df, index=c("store_id"),model= "random")	0.289
Purchasing Households	hhsrandom <- plm(log(purchasing_households) ~ circular + promotional_display + temp_price_reduction + price_product_charged + week_num + category + month + type_of_store, data=df, index=c("store_id"), model = "random")	0.284

Interpretation – PLM models

Variables	Coefficient	P-Value	Interpretation
Effect of a product being featured in the **circular** on **total spend**	0.758	(0.005)	A correlation matrix and plm model output show a strong positive correlation between a store having a promotional display for a product and the total amount spent on an order, meaning if a product is featured on the circular, the customer is more likely to spend more on their order.
Effect of a product having a **promotional display** on **total spend**	0.534	(0.005)	The correlation matrix and plm model output shows strong positive correlation between a store having a promotional display for a product and the total amount spent on an order, meaning if a promotional display is present, the customer is more likely to make a more expensive order.
Effect of a product having a **temporary price reduction** on **total spend**	-0.084	(0.004)	The correlation matrix and plm model output shows a slight negative correlation between a product having a temporary price reduction and the total amount spent on an order, meaning if a product has a temporary price reduction, the customer is likely to spend less on their order.
Effect of a product being featured in the **circular** on **the number of units sold**	0.727	(0.005)	The correlation matrix and plm model output shows strong positive correlation between a product being featured on the circular and the number of units that a customer purchases, meaning if a product is featured on the circular, the customer is more likely to purchase more units.
Effect of a product having a **promotional display** on **the number of units sold**	0.574	(0.005)	The correlation matrix and plm model output shows strong positive correlation between a store having a promotional display for a product and the number of units that a customer purchases, meaning if a promotional display is present, the customer is more likely to purchase more units.
Effect of a product having a **temporary price reduction** on **the number of units sold**	-0.039	(0.004)	The correlation matrix and plm model output shows slight negative correlation between a product having a temporary price reduction and the number of units that a customer purchases, meaning if a product has a temporary price reduction, the customer is more likely to purchase less units.
Effect of a product being featured in the **circular** on **the number of purchasing households**	0.679	(0.005)	The correlation matrix and plm model output shows strong positive correlation between a product being featured on the circular and the number household purchasers, meaning if a product is featured on the circular, there is likely to be more household purchasers of the product.
Effect of a product having a **promotional display** on **the number of purchasing households**	0.570	(0.005)	The correlation matrix and plm model output shows strong positive correlation between a product being featured on the circular and the number household purchasers, meaning if a promotional display is present,

			there is likely to be more household purchasers of the product.
Effect of a product having a **temporary price reduction** on **the number of purchasing households**	-0.090	(0.004)	The correlation matrix and plm model output shows very slight negative correlation between a product having a temporary price reduction and the number household purchasers, meaning if a product has a temporary price reduction, there is likely to be less household purchasers of the product.

Effects by Product Category and Store Segments

LMER Models

Variable	Model	R2 Value
Total Spend (Category)	lm19 <- lme4::lmer(log_total_spend ~ promotional_display + circular + temp_price_reduction + price_product_charged + week_num + month + category + (1 | store_id) , data=df , REML = FALSE)	**R2C: 0.4148**
Total Spend (Store Type)	lm22 = lme4::lmer(log_total_spend ~ promotional_display + circular + temp_price_reduction + price_product_charged + week_num + month + type_of_store + (1 | store_id) , data=df , REML = FALSE)	**R2C: 0.2223**
Units Sold (Category)	lm20 = lme4::lmer(log_units_sold ~ promotional_display + circular + temp_price_reduction + price_product_charged + week_num + month + category + (1 | store_id) , data=df , REML = FALSE)	**R2C: 0.4889**
Units Sold (Store Type)	lm23 = lme4::lmer(log_units_sold ~ promotional_display + circular + temp_price_reduction + price_product_charged + week_num + month + type_of_store + (1 | store_id) , data=df , REML = FALSE)	**R2C: 0.3525**

Purchasing Households (Category)	lm21 <- lme4::lmer(log_purchasing_households ~ promotional_display + circular + temp_price_reduction + price_product_charged + week_num + month + category + (1 | store_id) , data=df , REML = FALSE)	**R2C: 0.4984**
Purchasing Households (Store Type)	lm24 <- lme4::lmer(log_purchasing_households ~ promotional_display + circular + temp_price_reduction + price_product_charged + week_num + month + type_of_store + (1 | store_id) , data=df , REML = FALSE)	**R2C: 0.3537**

Interpretation – LMER model (Category Only)

Variables	Correlation Coefficient	P-Value	Interpretation
Effect of a product being featured in the **circular** on **total spend** **(Category)**	0.530	(0.005)	There is a positive correlation between log(total spend) and being featured on the circular. Compared to the PLM model, the product category of an item on display at a store has less impact on the total amount spent on an order than and item being on display in general.
Effect of a **product promotion** on **total spend** **(Category)**	-0.022	(0.004)	There is a slight negative correlation between log(total spend) and product promotion. Compared to the PLM model without the Category fixed effect, both models show that a price reduction has no impact (or slightly less) on the total spend of a customer order.
Effect of a product on **promotional display** on **total spend** **(Category)**	0.686	(0.004)	There is a positive correlation between log(Total Spend) and a product on promotional display. Compared to the PLM model without the Category fixed effect, the category of an item on display does provide a greater positive impact on the total spend of an order then grouping the items together.
Effect of a product being featured in the **circular** on **the number of units sold** **(Category)**	0.507	(0.005)	There is a positive correlation between log(Units Sold) and being featured on the circular. Compared to the PLM model, the category of item being featured in the circular has less of a positive effect on the number of units sold than an item in general being featured in the circular.

Effect of a **<u>product promotion</u>** on **<u>the number of units sold</u>** **(Category)**	0.012	(0.004)	There is a slight positive correlation between log(Units Sold) and product promotion. Compared to the PLM model, the category of an item has little impact on number units of sold for an order.
Effect of a product on **<u>promotional display</u>** on **<u>the number of units sold</u>** **(Category)**	0.708	(0.004)	There is a positive correlation between log(Units Sold) and a product on promotional display. Compared to the PLM Model, the category of and item on display has a greater impact in number of units sold versus grouping all items together.
Effect of a product being featured in the **<u>circular</u>** on **<u>the number of purchasing households</u>** **(Category)**	0.481	(0.005)	There is a positive correlation between log(Units Sold) and being featured in the circular. Compared to the PLM model, the category of an item being featured in the circular has less of an impact of number of purchasing households versus grouping all items together.
Effect of **<u>product promotion</u>** on **<u>the number of purchasing households</u>** **(Category)**	-0.025	(0.004)	There is a negative correlation between log(Purchasing Households) and promotions. Compared to the PLM model, the category of an item with a product promotion has little change of an impact of number of purchasing households versus grouping all items together.
Effect of a product on **<u>promotional display</u>** on **<u>the number of purchasing households</u>** **(Category)**	0.715	(0.004)	There is a positive correlation between log(Purchasing Households) and a product on promotional display. Compared to the PLM model, the category of an item on promotion display has a greater impact on number of purchasing households versus grouping all items together.

Interpretation – LMER model (Store Type Only)

Variables	Correlation Coefficient	P-Value	Interpretation
Effect of a product being featured in the **circular** on **total spend** **(Store Type)**	0.717	(0.005)	There is a positive correlation between log (total spend) and being featured on the circular. Compared to the PLM model, the type of store an item is being featured on the circular in has less impact on the total amount spent on an order than an item being on display in a store regardless of store type.
Effect of a **product promotion** on **total spend** **(Store Type)**	0.524	(0.005)	There is a positive correlation between log(total spend) and product promotion. Compared to the PLM model, the type of store a product promotion has greater positive impact on total spend then of then if an item were displayed in any store type.
Effect of a product on **promotional display** on **total spend** **(Store Type)**	-0.082	(0.004)	There is a light negative correlation between log(total spend) and a product on promotional display. Compared to the PLM model, the type of store a product is on promotional display in has a very slight negative effect on total spend compared to if an item was on promotional display in any store type.
Effect of a product being featured in the **circular** on **the number of units sold** **(Store Type)**	0.685	(0.005)	There is a positive correlation between log(Units Sold) and being featured on the circular. Compared to the PLM model, the type of store an item is being featured in the circular in has less of an impact on units sold then if an item was featured in the circular in any type of store.
Effect of a **product promotion** on **the number of units sold** **(Store Type)**	-0.037	(0.004)	There is a negative correlation between log(Units Sold) and product promotion. Compared to the PLM model, the type of store and item is on promotion in has no effect on number of units sold compared to if an item was on promotion in any store type.
Effect of a product on **promotional display** on **the number of units sold** **(Store Type)**	0.564	(0.005)	There is a positive correlation between log(Units Sold) and a product on promotional display. Compared to the PLM model, the type of store an item is promotional display in has nearly the same impact on units sold compared to an item being on promotional display in any store type.
Effect of a product being featured in the	0.640	(0.005)	There is a positive correlation between log(Purchasing Households) and being and promotional display. Compared to the PLM

circular on **the number of purchasing households** **(Store Type)**			model, the type of store an item is being featured in the circular in has slightly less impact on the number of purchasing households compared to an item being featured on the circular in any store type.
Effect of a **product promotion** on **the number of purchasing households** **(Store Type)**	-0.088	(0.004)	There is a negative correlation between log(Purchasing Households) and product promotion. Compared to the PLM model, the type of store a product an item is on promotion in has a slightly worse impact on the number of purchasing households for an item compared to the item being promotion in any story type.
Effect of a product on **promotional display** on **the number of purchasing households** **(Store Type)**	0.561	(0.005)	There is a positive correlation between log(Purchasing Households) and a product on promotional display. Compared to the PLM model, the type of store a product is on promotion display in has a worse impact on the number of purchasing households for an item compared to an item being on promotional display in any store type.

Price-Elasticity of Products

The elasticity of demand indicates how sensitive our customers are to price changes for a particular product so the product sales and units can be affected by price changes. To get coefficients for the formula for elasticity, we used a log-log linear model for each product in the dataset. To calculate the Elasticity, we use the formula:

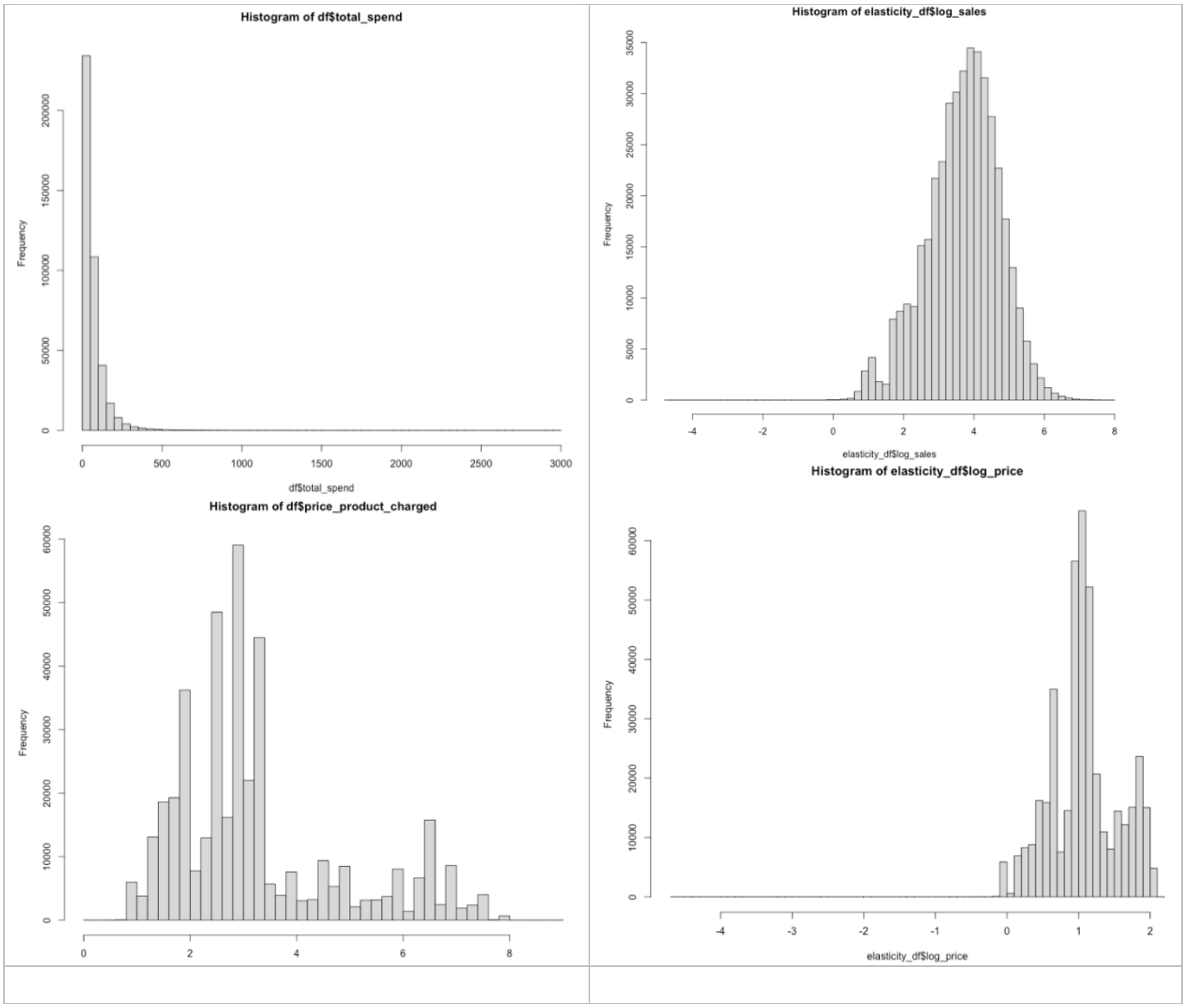
$$PE = \frac{\Delta Sales}{\Delta \mathrm{Pr}ice} * \frac{\mathrm{Pr}ice}{Sales}\text{ ; } having\ Sales\ as\ DV$$

$$PE = \frac{\Delta UnitsSold}{\Delta \mathrm{Pr}ice} * \frac{\mathrm{Pr}ice}{UnitsSold}\text{; } having\ Units\ Sold\ as\ DV$$

Where (ΔSales/ ΔPrice) and ΔUnitsSold/ ΔPrice) are given by the coefficient of each regression model, and Price, Sales, and Unit Sold are determined using mean (Sales), mean (Price), and mean(Units old) in R, respectively. Total spend has been renamed to Sales for a better understanding of the interpretation.

"total_spend", "units_sold", and "price_product_charged" in this dataset were transformed into logarithmic values due to their dispersion. Below is a figure showing how the dependent variables differ before and after a logarithmic transformation.

Histograms



Model

The model and elasticity can be obtained in R as follows:

```
model_for_elasticity = lm(log_sales ~ log_price,
data=only_one_product_at_a_time) #Sales as DV

PE = as.numeric(model_for_elasticity$coefficients["log_price"] *
mean(only_one_product_at_a_time$log_price)/mean(only_one_product_at_a_time
$log_sales)) #Sales as DV

model_for_elasticity = lm(log_units ~ log_price,
data=only_one_product_at_a_time)#Units as DV

PE = as.numeric(model_for_elasticity$coefficients["log_price"] *
mean(only_one_product_at_a_time$log_price)/mean(only_one_product_at_a_time
$log_units)) #Units as DV
```

A for loop has been used to create a temporal subset of data for each product in order to calculate its elasticity. The results have been introduced into a new dataset called “elasticity_by_product” using the rbind() function (see R code part 3 for reference).

```
list_of_products = unique(elasticity_df$product_name)
for (i in list_of_products){
  elasticity_by_product = rbind(elasticity_by_product,calc_elasticity(i))
}
```

Due to a log transformation of price, units sold, and sales transform units on demand into the percentage change in demand resulting from a percentage change in price.
Using sales and units sold as dependent variables, the top 5 elastic and inelastic products are generated after elasticity is calculated for each product.

Top 5 Elastic and inelastic products by sales as DV

```
> top_five_elastic
             product_name elasticity_num     category direction
1      FRSC PEPPERONI PIZZA           1.69 FROZEN PIZZA       [-]
2 FRSC BRCK OVN ITL PEP PZ           1.53 FROZEN PIZZA       [-]
3       DIGIORNO THREE MEAT           1.39 FROZEN PIZZA       [-]
4       FRSC 4 CHEESE PIZZA           1.34 FROZEN PIZZA       [-]
5    NWMN OWN SUPREME PIZZA           1.27 FROZEN PIZZA       [-]
> tope_five_inelastic
             product_name elasticity_num    category direction
1    SHURGD PRETZEL STICKS           0.08  BAG SNACKS       [+]
2           PL RAISIN BRAN           0.08 COLD CEREAL       [+]
3      SHURGD PRETZEL RODS           0.06  BAG SNACKS       [+]
4 PL BT SZ FRSTD SHRD WHT           0.03 COLD CEREAL       [+]
5              GM CHEERIOS           0.01 COLD CEREAL       [-]
```

Top 5 Elastic and inelastic products by Units as DV

```
> top_five_elastic_units
             product_name elasticity_num     category direction
1      FRSC PEPPERONI PIZZA           5.12 FROZEN PIZZA       [-]
2 FRSC BRCK OVN ITL PEP PZ           4.53 FROZEN PIZZA       [-]
3       FRSC 4 CHEESE PIZZA           4.47 FROZEN PIZZA       [-]
4    NWMN OWN SUPREME PIZZA           4.43 FROZEN PIZZA       [-]
5 NWMN OWN PEPPERONI PIZZA           3.60 FROZEN PIZZA       [-]
> tope_five_inelastic_units
             product_name elasticity_num    category direction
1      SHURGD MINI PRETZELS           0.10  BAG SNACKS       [-]
2     SHURGD PRETZEL STICKS           0.10  BAG SNACKS       [-]
3         PL PRETZEL STICKS           0.08  BAG SNACKS       [+]
4            PL RAISIN BRAN           0.07 COLD CEREAL       [-]
5 PL HONEY NUT TOASTD OATS           0.04 COLD CEREAL       [+]
```

Interpretation

Considering a hypothetical product with an elasticity equal to 0 is perfectly inelastic, meaning a price increase will not affect sales; therefore, based on the results, the Manager should increase the price of the top five inelastic products to maximize product sales, since a price increase will not have a significant impact on sales. In contrast, he should also consider decreasing the price of the top five elastic products to maximize unit sales, since a ten percent price reduction will result in a 5.12% increase in unit sales, for example on FRSC PEPPERONI PIZZA.

Lessons Learned & Course Feedback

Lessons Learned

As a whole, this class has demonstrated how powerful statistics can be in predicting real-world models. As a result of out-of-class projects, the ability to analyze dependent and independent variables using the multicollinearity test has increased, as well as the proficiency in the use of tools provided to verify model assumptions, heteroskedasticity, and homoskedasticity for non-linear models.

Course Feedback

Overall, the course was challenging, the professor did a good job at the start of the course refreshing on key concepts from QMB 6304. The material was challenging (and most of it) went over

Appendix

Stargazer Output for PLM Models

	Dependent variable:		
	log(total_spend)	log(units_sold)	log(purchasing_households)
	(1)	(2)	(3)
circular	0.758***	0.727***	0.679***
	(0.005)	(0.005)	(0.005)
promotional_display	0.534***	0.574***	0.570***
	(0.005)	(0.005)	(0.005)
temp_price_reduction	-0.084***	-0.039***	-0.090***
	(0.004)	(0.004)	(0.004)
price_product_charged	-0.0002	-0.280***	-0.271***
	(0.001)	(0.001)	(0.001)
week_num	-0.001***	-0.001***	-0.001***
	(0.00003)	(0.00003)	(0.00003)
monthApr	-0.052***	-0.050***	-0.053***
	(0.007)	(0.007)	(0.007)
monthAug	-0.045***	-0.043***	-0.042***
	(0.007)	(0.007)	(0.007)
monthDec	0.033***	0.038***	0.024***
	(0.007)	(0.007)	(0.007)
monthFeb	0.007	0.010	0.009
	(0.007)	(0.007)	(0.007)
monthJul	-0.064***	-0.076***	-0.073***
	(0.007)	(0.007)	(0.007)
monthJun	-0.041***	-0.048***	-0.046***
	(0.007)	(0.007)	(0.007)
monthMar	-0.097***	-0.108***	-0.108***
	(0.007)	(0.007)	(0.007)
monthMay	-0.039***	-0.041***	-0.044***
	(0.007)	(0.007)	(0.007)
monthNov	-0.025***	-0.028***	-0.037***
	(0.007)	(0.007)	(0.007)
monthOct	-0.003	-0.017**	-0.021***
	(0.007)	(0.007)	(0.007)
monthSep	-0.030***	-0.036***	-0.034***
	(0.007)	(0.007)	(0.007)
Constant	3.617***	3.457***	3.300***
	(0.023)	(0.023)	(0.023)
Observations	418,554	418,554	418,554
R^2	0.126	0.289	0.284

Adjusted R^2	0.126	0.289	0.284
F Statistic	60,633.110***	170,437.300***	166,321.500***
Note:			*p**p***p<0.01

Stargazer Output For Store Type Only Model

	Dependent variable:		
	log_total_spend	log_units_sold	log_purchasing_households
	(1)	(2)	(3)
promotional_displayYes	0.524***	0.564***	0.561***
	(0.005)	(0.005)	(0.005)
circularYes	0.717***	0.685***	0.640***
	(0.005)	(0.005)	(0.005)
temp_price_reductionYes	-0.082***	-0.037***	-0.088***
	(0.004)	(0.004)	(0.004)
price_product_charged	0.002*	-0.279***	-0.270***
	(0.001)	(0.001)	(0.001)
week_num	-0.001***	-0.001***	-0.001***
	(0.00003)	(0.00003)	(0.00003)
monthApr	-0.055***	-0.053***	-0.056***
	(0.007)	(0.007)	(0.007)
monthAug	-0.053***	-0.052***	-0.050***
	(0.007)	(0.007)	(0.007)
monthDec	0.025***	0.030***	0.016**
	(0.007)	(0.007)	(0.007)
monthFeb	-0.004	-0.001	-0.001
	(0.007)	(0.007)	(0.007)
monthJul	-0.068***	-0.080***	-0.077***
	(0.007)	(0.007)	(0.007)
monthJun	-0.044***	-0.050***	-0.049***
	(0.007)	(0.007)	(0.007)
monthMar	-0.101***	-0.112***	-0.111***
	(0.007)	(0.007)	(0.007)
monthMay	-0.043***	-0.045***	-0.048***
	(0.007)	(0.007)	(0.007)
monthNov	-0.032***	-0.034***	-0.042***
	(0.007)	(0.007)	(0.007)
monthOct	-0.005	-0.019***	-0.022***
	(0.007)	(0.007)	(0.007)
monthSep	-0.033***	-0.040***	-0.037***
	(0.007)	(0.007)	(0.007)
type_of_storeMAINSTREAM	0.383***	0.385***	0.419***

	(0.083)	(0.082)	(0.082)
type_of_storeUPSCALE	0.388***	0.389***	0.423***
	(0.083)	(0.082)	(0.082)
Constant	3.326***	3.165***	2.983***
	(0.072)	(0.071)	(0.071)
Observations	417,188	417,188	417,188
Log Likelihood	-545,587.600	-540,531.400	-531,086.600
Akaike Inf. Crit.	1,091,217.000	1,081,105.000	1,062,215.000
Bayesian Inf. Crit.	1,091,447.000	1,081,335.000	1,062,445.000
Note:			*p**p***p<0.01

Stargazer Output for Category Only model

	Dependent variable:		
	log_total_spend	log_units_sold	log_purchasing_households
	(1)	(2)	(3)
promotional_displayYes	0.686***	0.708***	0.715***
	(0.004)	(0.004)	(0.004)
circularYes	0.530***	0.507***	0.481***
	(0.005)	(0.005)	(0.005)
temp_price_reductionYes	-0.022***	0.012***	-0.025***
	(0.004)	(0.004)	(0.004)
price_product_charged	-0.016***	-0.304***	-0.272***
	(0.001)	(0.001)	(0.001)
week_num	-0.001***	-0.001***	-0.001***
	(0.00003)	(0.00003)	(0.00003)
monthApr	-0.055***	-0.052***	-0.056***
	(0.006)	(0.006)	(0.006)
monthAug	-0.057***	-0.055***	-0.054***
	(0.006)	(0.006)	(0.006)
monthDec	0.024***	0.029***	0.015***
	(0.006)	(0.006)	(0.006)
monthFeb	-0.010	-0.007	-0.006
	(0.006)	(0.006)	(0.006)
monthJul	-0.078***	-0.089***	-0.086***
	(0.006)	(0.006)	(0.006)
monthJun	-0.053***	-0.058***	-0.058***
	(0.006)	(0.006)	(0.006)
monthMar	-0.104***	-0.114***	-0.113***
	(0.006)	(0.006)	(0.006)
monthMay	-0.046***	-0.047***	-0.050***
	(0.006)	(0.006)	(0.006)

monthNov	-0.026***	-0.030***	-0.037***
	(0.006)	(0.006)	(0.006)
monthOct	-0.013**	-0.027***	-0.029***
	(0.006)	(0.006)	(0.006)
monthSep	-0.048***	-0.053***	-0.051***
	(0.006)	(0.006)	(0.006)
categoryCOLD CEREAL	1.023***	0.936***	0.928***
	(0.003)	(0.003)	(0.003)
categoryFROZEN PIZZA	0.508***	0.508***	0.400***
	(0.005)	(0.005)	(0.005)
Constant	3.090***	2.996***	2.791***
	(0.045)	(0.044)	(0.045)
Observations	417,188	417,188	417,188
Log Likelihood	-492,445.000	-496,565.700	-483,755.600
Akaike Inf. Crit.	984,932.000	993,173.300	967,553.200
Bayesian Inf. Crit.	985,161.800	993,403.100	967,783.000
Note:			*p**p***p<0.01

Full R Code

```
rm(list=ls())
library(rio)

#Import SnackChain.Xlsx file.
stores_df       <- import("SnackChain.xlsx", sheet="stores")  #79 stores
products_df     <- import("SnackChain.xlsx", sheet="products") #58
products
transactions_df <- import("SnackChain.xlsx", sheet="transactions") #524950
transactions
#Joining Tables.
temp <- merge(transactions_df, products_df, by.x=c("UPC"), by.y=c("UPC"))
#join by UPCs between transactions and products
df   <- merge(temp, stores_df, by.x=c("STORE_NUM"), by.y=c("STORE_ID"))
#join by adding the stores.
#Removing temporary Files.
rm(transactions_df)
rm(stores_df)
rm(temp)
rm(products_df)
#___________________________________________________________________________
                            #Data Preprocessing
#___________________________________________________________________________
#Remove oral hygiene products
df <- df[df$CATEGORY != "ORAL HYGIENE PRODUCTS", ]  #this could be for
another project

#Check for missing values
```

```
colSums(is.na(df))
#' 282548 missing PARKING values; we will not use this data since it is
higher (store) level,
#' 10 NA values for price and 173 for BASE_PRICE
df <- subset(df, select = -c(PARKING))
df <- df[complete.cases(df), ] #Removing missing values

#Renaming columns to make them easier to understand
colnames(df)[colnames(df) == "STORE_NUM"] = "store_id"
colnames(df)[colnames(df) == "UPC"] = "product_id"
colnames(df)[colnames(df) == "WEEK_END_DATE"] = "week_ending_date"
colnames(df)[colnames(df) == "UNITS"] = "units_sold"
colnames(df)[colnames(df) == "VISITS"] = "unique_purchases"
colnames(df)[colnames(df) == "HHS"] = "purchasing_households"
colnames(df)[colnames(df) == "SPEND"] = "total_spend"
colnames(df)[colnames(df) == "PRICE"] = "price_product_charged"
colnames(df)[colnames(df) == "BASE_PRICE"] = "base_price_product"
colnames(df)[colnames(df) == "FEATURE"] = "circular"
colnames(df)[colnames(df) == "DISPLAY"] = "promotional_display"
colnames(df)[colnames(df) == "TPR_ONLY"] = "temp_price_reduction"
colnames(df)[colnames(df) == "DESCRIPTION"] = "product"
colnames(df)[colnames(df) == "MANUFACTURER"] = "manufacturer"
colnames(df)[colnames(df) == "CATEGORY"] = "category"
colnames(df)[colnames(df) == "SUB_CATEGORY"] = "subcategory"
colnames(df)[colnames(df) == "PRODUCT_SIZE"] = "size"
colnames(df)[colnames(df) == "STORE_NAME"] = "store_name"
colnames(df)[colnames(df) == "CITY"] = "city"
colnames(df)[colnames(df) == "STATE"] = "state"
colnames(df)[colnames(df) == "MSA"] = "metropilitan_area"
colnames(df)[colnames(df) == "SEGMENT"] = "type_of_store"
colnames(df)[colnames(df) == "SIZE"] = "store_sqft"
colnames(df)[colnames(df) == "AVG_WEEKLY_BASKETS"] =
"avg_weekly_baskets_sold"

#Converting Categorical variables into factor variables.
df$store_id = factor(df$store_id)
df$product_id = factor(df$product_id)
df$category = factor(df$category)
df$subcategory = factor(df$subcategory)
df$category = relevel(df$category, "BAG SNACKS")
df$city = factor(df$city)
df$state = factor(df$state)
df$type_of_store = factor(df$type_of_store)
df$type_of_store = relevel(df$type_of_store, "VALUE")
df$product = factor(df$product)


#Extracting year, month, and week_ending_date from date.
df$year = format(df$week_ending_date, "%Y")
df$year = factor(df$year)
df$month = format(df$week_ending_date, "%b")
df$month = factor(df$month)
df$month = relevel(df$month, "Jan")
```

```
df$week_num = difftime(df$week_ending_date, df$week_ending_date[1],
units="weeks")
df$week_num = df$week_num + 1
attach(df)
str(df)

df = df[df$price_product_charged != 0, ] #eliminating 0's on price product
to avoid errors when performing log()
df = df[df$total_spend != 0, ] #eliminating 0's on total spend

df$log_total_spend <- log(df$total_spend)
df$log_units_sold <- log(df$units_sold)
df$log_purchasing_households <- log(df$purchasing_households)
df$log_price_product_charged <- log(df$price_product_charged)

df <- subset(df, df$log_total_spend != -Inf)
#_________________________________________________________________________
#    1. Effect of promotional displays or being featured in the circular
#_________________________________________________________________________
#Data Visualization
hist(df$total_spend)
hist(log(df$total_spend))
hist(df$units_sold)
hist(log(df$units_sold))
hist(df$purchasing_households)
hist(log(df$purchasing_households))

#Correlations
library(corrplot)
correlation = cor(df[c("units_sold", "promotional_display",
"purchasing_households", "total_spend", "temp_price_reduction",
                       "circular", "price_product_charged",
"base_price_product", "metropilitan_area",
                       "store_sqft", "avg_weekly_baskets_sold")],
use="pairwise.complete.obs")
corrplot(correlation)

#Plots
boxplot(df$total_spend ~ df$promotional_display)
boxplot(df$total_spend ~ df$circular)
boxplot(df$total_spend ~ df$city)
boxplot(df$total_spend ~ df$temp_price_reduction)
boxplot(df$total_spend ~ df$type_of_store)

library(car)
library(lmtest)
library(plm)


# Random Effect Model for spend
spendrandom <- plm(log(total_spend) ~ circular
                   + promotional_display
                   + temp_price_reduction
                   + price_product_charged
```

```
                   + week_num
                   + month,
                   data=df, index=c("store_id"),model= "random")

# Random Effect Model for spend
unitsrandom <- plm(log(units_sold) ~ circular
                   + promotional_display
                   + temp_price_reduction
                   + price_product_charged
                   + week_num
                   + month,
                   data=df, index=c("store_id"),model= "random")

# Random Effect Model for spend
hhsrandom <- plm(log(purchasing_households) ~ circular
                 + promotional_display
                 + temp_price_reduction
                 + price_product_charged
                 + week_num
                 + month,
                 data=df, index=c("store_id"),model= "random")

library(stargazer)
stargazer(spendrandom, unitsrandom, hhsrandom, type = "html",
out="~/Downloads/SnackChain.html")
#__________________________________________________________________________
#               2. Models w/out category and type of store
#__________________________________________________________________________
#Best models from question 1 to be used as the base for question 2

#Total Spend  base model
lm10 = lm(log_total_spend ~  promotional_display + circular +
temp_price_reduction + store_id + week_num + month , data=df)
summary(lm10 )
par(mfrow=c(2,2))
plot(lm10 )


#Random Effect Week Number
lm11 = lme4::lmer(log_total_spend  ~ promotional_display + circular +
temp_price_reduction + week_num + month + store_id + (1 | week_num),
data=df, REML = FALSE)
summary(lm11)
par(mfrow=c(2,2))
plot(lm11)

qqnorm(resid(lm10))
qqline(resid(lm10))

#Random Effect Week Number and Store Number
lm12 = lme4::lmer(log_total_spend ~ promotional_display + circular +
temp_price_reduction + price_product_charged  + week_num + month  + (1 |
store_id) , data=df, REML = FALSE)
summary(lm12)
```

```
par(mfrow=c(2,2))
plot(lm12)

qqnorm(resid(lm11))
qqline(resid(lm11))


library('MuMIn')

r.squaredGLMM(lm11)
r.squaredGLMM(lm12)


library(stargazer)
stargazer(lm10, lm11, lm12, type="html", out="totalspend.html")
AIC(lm10, lm11, lm12)



#Units Sold base model
lm13 = lm(log_units_sold ~ promotional_display + circular +
temp_price_reduction + store_id + week_num + month + category +
type_of_store  , data=df)
summary(lm13)
lm13$coefficients
par(mfrow=c(2,2))
plot(lm13)


#Random Effect Week Number
lm14 = lme4::lmer(log_units_sold ~ promotional_display + circular +
temp_price_reduction + store_id + week_num + month + category +
type_of_store + (1 | week_num), data=df, REML = FALSE)
summary(lm14)
par(mfrow=c(2,2))
plot(lm14)

qqnorm(resid(lm14))
qqline(resid(lm14))

#Random Effect Week Number and Store Number
lm15 = lme4::lmer(log_units_sold ~ promotional_display + circular +
temp_price_reduction + price_product_charged  + week_num + month +
category + type_of_store  + (1 | store_id) , data=df, REML = FALSE)
summary(lm15)
par(mfrow=c(2,2))
plot(lm15)

qqnorm(resid(lm15))
qqline(resid(lm15))


library('MuMIn')
```

```
r.squaredGLMM(lm14)
r.squaredGLMM(lm15)


library(stargazer)
stargazer(lm13, lm14, lm15, type="html", out="unitssold.html")
AIC(lm13, lm14, lm15)




#Purchasing household  base model
lm16 = lm(log_purchasing_households ~ promotional_display + circular +
temp_price_reduction + store_id + week_num + month  , data=df)
summary(lm16)
lm16$coefficients
par(mfrow=c(2,2))
plot(lm16)



#Random Effect Week Number
lm17 = lme4::lmer(log_purchasing_households ~ promotional_display +
circular + temp_price_reduction + store_id + month + (1 | week_num),
data=df, REML = FALSE)
summary(lm17)
par(mfrow=c(2,2))
plot(lm17)

qqnorm(resid(lm17))
qqline(resid(lm17))

#Random Effect Week Number and Store Number
lm18 <- lme4::lmer(log_purchasing_households  ~ promotional_display +
circular + temp_price_reduction + price_product_charged  + week_num +
month + (1 | store_id) , data=df, REML = FALSE)
summary(lm18)
par(mfrow=c(2,2))
plot(lm18)

qqnorm(resid(lm18))
qqline(resid(lm18))

install.packages('MuMIn')
library('MuMIn')

r.squaredGLMM(lm17)
r.squaredGLMM(lm18)

library(stargazer)
stargazer(lm16, lm17, lm18, type="html", out="PurchasingHouseholds.html")
AIC(lm16, lm17, lm18)
```

```
df_CC <- df[df$category == "COLD CEREAL", ]
df_FP <- df[df$category == "FROZEN PIZZA", ]
df_MS  <- df[df$type_of_store == "MAINSTREAM", ]
df_UP  <- df[df$type_of_store == "UPSCALE", ]


#__________________________________________________________________________
#                            2. Category Only
#__________________________________________________________________________
#Random Effect Week Number and Store Number
lm19 <- lme4::lmer(log_total_spend ~ promotional_display
                   + circular
                   + temp_price_reduction
                   + price_product_charged
                   + week_num
                   + month
                   + category
                   + (1 | store_id) , data=df , REML = FALSE)
summary(lm19)
par(mfrow=c(2,2))
plot(lm19)



#Random Effect Week Number and Store Number
lm20 = lme4::lmer(log_units_sold ~ promotional_display
                  + circular
                  + temp_price_reduction
                  + price_product_charged
                  + week_num
                  + month
                  + category
                  + (1 | store_id) , data=df , REML = FALSE)
summary(lm20)
par(mfrow=c(2,2))
plot(lm20)


#Random Effect Week Number and Store Number
lm21 <- lme4::lmer(log_purchasing_households  ~ promotional_display
                   + circular
                   + temp_price_reduction
                   + price_product_charged
                   + week_num
                   + month
                   + category
                   + (1 | store_id) , data=df , REML = FALSE)
summary(lm21)
par(mfrow=c(2,2))
plot(lm21)

library('MuMIn')
r.squaredGLMM(lm19)
r.squaredGLMM(lm20)
```

```
r.squaredGLMM(lm21)


library(stargazer)
stargazer(lm19, lm20, lm21, type="html", out="category.html")

#_____________________________________________________________________________
#                             2. Type of Store Only
#______________________________________________________________________________
#Random Effect Week Number and Store Number
lm22 = lme4::lmer(log_total_spend ~ promotional_display
                  + circular
                  + temp_price_reduction
                  + price_product_charged
                  + week_num
                  + month
                  + type_of_store  + (1 | store_id) , data=df , REML =
FALSE)
summary(lm22)
par(mfrow=c(2,2))
plot(lm22)

#Random Effect Week Number and Store Number
lm23 = lme4::lmer(log_units_sold ~ promotional_display
                  + circular
                  + temp_price_reduction
                  + price_product_charged
                  + week_num
                  + month
                  + type_of_store
                  + (1 | store_id) , data=df , REML = FALSE)
summary(lm23)
par(mfrow=c(2,2))
plot(lm23)

#Random Effect Week Number and Store Number
lm24 <- lme4::lmer(log_purchasing_households  ~ promotional_display
                   + circular
                   + temp_price_reduction
                   + price_product_charged
                   + week_num
                   + month
                   + type_of_store
                   + (1 | store_id) , data=df , REML = FALSE)
summary(lm24)
par(mfrow=c(2,2))
plot(lm24)|

  library('MuMIn')
r.squaredGLMM(lm22)
r.squaredGLMM(lm23)
r.squaredGLMM(lm24)

library(stargazer)
```

```
stargazer(lm22, lm23, lm24, type="html", out="type_of_store.html")
#_________________________________________________________________________
#                            3. Price Elasticity
#_________________________________________________________________________
#Data exploration
table(product)
table(product_id)
table(category)
summary(df)
hist(df$price_product_charged, breaks = 50)
hist(df$total_spend, breaks = 50)
df = df[df$units_sold != 1800, ]
hist(log(df$units_sold), breaks = 30)
#summary(df)

#Creating a dataframe to store log(sales), log(price), category, and
product name separately to avoid row disparity.
sales = as.vector(df[,'total_spend'])
units = as.vector(df[, 'units_sold'])
price = as.vector(df[,'price_product_charged'])
category = as.vector(df[,'category'])
product_name = as.vector(df[,'product'])
households = as.vector(df[,'purchasing_households'])
elasticity_df = data.frame(log(sales), log(price), log(units),
log(households),category, product_name)
colnames(elasticity_df) = c("log_sales", "log_price", "log_units",
"log_purchasing_households","category", "product_name")
hist(elasticity_df$log_sales, breaks = 50)
hist(elasticity_df$log_price, breaks = 50)
hist(elasticity_df$log_units, breaks = 50)
rm(sales, units, price, category, product_name)
#summary(elasticity_df)

#Creating a function that calculates the elasticity given a dependent
variable and a product name (product name can be iterated from a list).
#The function returns the product name, elasticity and its category. PE =
Price Elasticity
elasticity_by_product = data.frame(product_name=c("Perfectly Inelastic
Test product name"), elasticity_num=c(0),  category=c("Test_name"))
calc_elasticity = function(x){
  only_one_product_at_a_time = subset(elasticity_df, product_name==x)
  model_for_elasticity = lm(log_sales ~ log_price, data =
only_one_product_at_a_time)
  PE = as.numeric(model_for_elasticity$coefficients["log_price"] *
mean(only_one_product_at_a_time$log_price)/mean(only_one_product_at_a_time
$log_sales))
  category_name = (only_one_product_at_a_time$category[1])
  aux_vector = c(x, PE, category_name)
  return(aux_vector)
}

#Iterating and adding each product with its respective PE to
elasticity_by_product dataframe.
```

```
list_of_products = unique(elasticity_df$product_name) # Making a list of
the 41 unique products.
for (i in list_of_products){
  elasticity_by_product = rbind(elasticity_by_product, calc_elasticity(i))
}


#Data transformation on elasticity_by_product dataset.
elasticity_by_product$elasticity_num =
as.numeric(elasticity_by_product$elasticity_num) #Converting to Number
elasticity_by_product <- elasticity_by_product[-1,] #Deleting the test
product with elasticity = 0
library(dplyr)
elasticity_by_product <- elasticity_by_product %>%
  mutate(direction = if_else(elasticity_num > 0, "[+]", "[-]"))#Preserving
the symbol before to perform abs()
library(tidyverse)
elasticity_by_product$elasticity_num =
abs(elasticity_by_product$elasticity_num) #Making absolute values
elasticity_by_product$elasticity_num =
round(elasticity_by_product$elasticity_num, digits = 2)
elasticity_by_product = elasticity_by_product[order(-
elasticity_by_product$elasticity_num),]#Sorting data frame

#Filtering the Top 5, most and least elastic products:
top_five_elastic = elasticity_by_product[1:5,]
row.names(top_five_elastic) <- NULL
tope_five_inelastic = tail(elasticity_by_product, n = 5)
row.names(tope_five_inelastic) <- NULL
top_five_elastic
tope_five_inelastic

     #Analysis of elasticity using units_sold as a dependent variable.
elasticity_by_product_units = data.frame(product_name=c("Perfectly
Inelastic Test product name"), elasticity_num=c(0),
category=c("Test_name"))
calc_elasticity_units = function(x){
  only_one_product_at_a_time = subset(elasticity_df, product_name==x)
  model_for_elasticity = lm(log_units ~ log_price, data =
only_one_product_at_a_time)
  PE = as.numeric(model_for_elasticity$coefficients["log_price"] *
mean(only_one_product_at_a_time$log_price)/mean(only_one_product_at_a_time
$log_units))
  category_name = (only_one_product_at_a_time$category[1])
  aux_vector = c(x, PE, category_name)
  return(aux_vector)
}
for (i in list_of_products){
  elasticity_by_product_units = rbind(elasticity_by_product_units,
calc_elasticity_units(i))
}
elasticity_by_product_units$elasticity_num =
as.numeric(elasticity_by_product_units$elasticity_num) #Converting to
Number
```

```
elasticity_by_product_units <- elasticity_by_product_units[-1,] #Deleting
the test product with elasticity = 0
elasticity_by_product_units <- elasticity_by_product_units %>%
  mutate(direction = if_else(elasticity_num > 0, "[+]", "[-]"))#Preserving
the symbol before to perform abs()
elasticity_by_product_units$elasticity_num =
abs(elasticity_by_product_units$elasticity_num) #Making absolute values
elasticity_by_product_units$elasticity_num =
round(elasticity_by_product_units$elasticity_num, digits = 2)
elasticity_by_product_units = elasticity_by_product_units[order(-
elasticity_by_product_units$elasticity_num),]#Sorting data frame

#Top 5, most and least elastic products for units sold:
top_five_elastic_units = elasticity_by_product_units[1:5,]
row.names(top_five_elastic_units) <- NULL
tope_five_inelastic_units = tail(elasticity_by_product_units, n = 5)
row.names(tope_five_inelastic_units) <- NULL
top_five_elastic_units
tope_five_inelastic_units


#Determining the LINE assumptions of regression for one product.
assumptions_LINE = function(product_name) {
  only_one_product = subset(df, product==product_name)
  m1 = lm(log_total_spend ~ log_price_product_charged +
log_purchasing_households, data=only_one_product)
  #m1 = lm(total_spend ~ price_product_charged + units_sold +
purchasing_households, data=only_one_product)
  par(mfrow=c(2,2))
  # Linearity
  plot(only_one_product$total_spend,m1$fitted.values, pch=19,main="Actuals
v. Fitteds, total_spend")
  abline(0,1,col="red",lwd=3)
  # Normality
  qqnorm(m1$residuals,pch=19,
         main="Normality Plot, total_spend")
  qqline(m1$residuals,lwd=3,col="red")
  hist(m1$residuals,col="red",
       main="Residuals, total_spend",
       probability=TRUE)
  curve(dnorm(x,mean(m1$residuals),
              sd(m1$residuals)),
        from=min(m1$residuals),
        to=max(m1$residuals),
        lwd=3,col="Black",add=TRUE)
  # Equality of Variances
  plot(m1$fitted.values,rstandard(m1),
       pch=19,main="Equality of Variances, total_spend")
  abline(0,0,lwd=3,col="red")
  par(mfrow=c(1,1))
  summary(m1)
}

#assumptions_LINE("FRSC PEPPERONI PIZZA")
```