Big Data Integration Final Paper

Sidney Schaeper, Aman Rastogi, Keya Satpathy, Bhawna Saini, Prathiba Swamykannu

2/20/2020

### Understanding Our Topic

We had a variety of questions we wanted to answer in this project. We formualted these questions after evalauting our 84.51 datasets and the twitter data. The following are the questions we hoped to answer.

* Should we continue to do business with our five worst performing brands based on the perspective of customers?
* Does location of the product within the store and ad impact sales of the product? What is the optimal location within the store and ads?
* What amounts of each product should we plan to have in inventory on average daily based on these sales?

For the first question, we thought that the twitter dataset, product\_lookup dataset, and transactions dataset would provide us insight into answering this question. For the second question, we thought that the transactions dataset and causal\_lookup dataset would provide us information into answering this question. For the third question, we thought that the transactions dataset, store\_lookup dataset, and product\_lookup dataset would help us discover the answer to this question.

We thought the first question would be insightful to the business, because a company should know what customers think of the brands they partner with. A company likely wouldn’t want to work with a brand that has a bad image to its customers, because they likely wouldn’t sell many of the products associated with that brand. In addition, we considered the second question to be a great question to answer, because we thought this could help Kroger’s marketing strategy. The marketing strategy would be improved with this knowledge, because the marketers would have more knowledge of the optimal locations for high product sales in the weekly ad and stores. The final question would be great for a busines to know, because this knowledge would help improve a company’s inventory strategy. This strategy would improve, because you possibly could improve your inventory prediction levels for each store from using the data instead of just guessing.

### Describe The Tools Used

Tools that we used were basically R packages which include resuable R functions. They come in really handy for complex data analysis as ours.

The packages that we used are:  
1. **readr** - to read files into R  
2. **dplyr** - for data wrangling and manipulation  
3. **tidyr** - for data cleaning  
4. **shiny** - for creating interactive reports  
5. **DT** - to include DataTables  
6. **ggplot2** - for fancy visualizations  
7. **plotly** - for interactive plots and graphs

#Packages that needed to be installed for this project#  
install.packages("readr")  
install.packages("dplyr")  
install.packages("tidyr")  
install.packages("shiny")  
install.packages("DT")  
install.packages("ggplot2")  
install.packages("plotly")  
install.packages("tidyverse")  
install.packages("tm")  
install.packages("ggwordcloud")  
install.packages("tidyr")  
install.packets("rtweet")

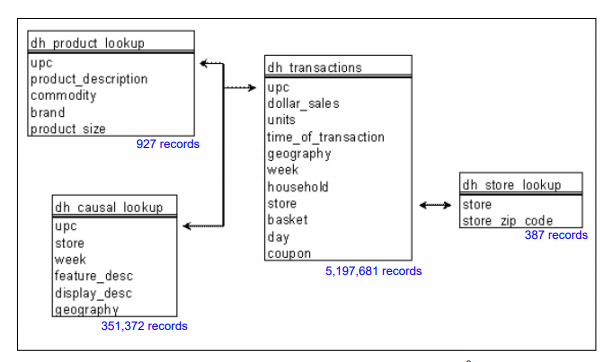
#Packages that we needed for the presentation but can't have in the paper due to these being interactive#  
library(shiny)  
library(DT)

#Packages that need to be found for this project for the paper#  
library(readr)  
library(dplyr)  
library(tidyr)  
library(ggplot2)  
library(plotly)  
library(tidyverse)  
library(tm)  
library(ggwordcloud)  
library(tidyr)  
library(rtweet)

### 84.51 Data Source

The data that we wanted to evaluate for this project came from two sources. The first source was from [84.51’s website](https://www.8451.com/). The folder of data that we used from 84.51 are called the Carbo Loading datasets. There are a total of four datasets within this folder. The four datasets are called the causal\_lookup, product\_lookup, store\_lookup, and transactions. According to the Carbo Loading guide provided, these datasets came from a relational database. Also, these datasets contain purchases at a household level over two years, and the datasets are filtered to only contain products from four categories. These four categories are pasta, pasta sauce, syrup, and pancake mix. In addition, the guide contained information pertaining to the variables within each of the datasets. The tables below show these variables and their description.

The pasta, pasta sauce, syrup, and pancake mix household level transaction data was obtained through a loyalty card program of a leading US grocer. These transactions were monitored over two years and a total of over 5 million specific product purchases. These 5 million specific product purchases were documented across 387 unique stores of the leading US grocer. A total of 927 different products within the four commodities were recorded during this period. Furthermore, each product’s location within a specific weekly mailer was documented and tracked over the monitored two year period.

Below is the overall view of the table: 

**Transactions Dataset Table** For the entire two year period in our data, 2500 frequent shoppers were tracked on the basis of the grocery transactions. Every single purchase is accounted for,on every single shopping trip. Price information, coupon use information is readily available. The quantity of a specific product is also measured.

|  |  |
| --- | --- |
| Variable | Description |
| upc | It is a standard 10 digit code assigned to products. This is the product that was purchased. |
| dollar\_sales | The amount of money spent on this product by the customer. These are recorded in dollars. |
| units | The quantity of this product purchased. |
| time\_of\_transaction | The time the transaction occurred. This is recorded in military time. |
| geography | This label tells you where it was purchased out of the two large regions. These two regions consist of multiple states. The value can either be 1 or 2. |
| week | This notifies the week that the transaction occured. The range of values is from 1 to 104. These numbers are assigned chronologically. |
| household | This value is a unique number assigned to a household. This is the purchaser of the product. |
| store | This value is a unique number assigned to each store. This is where the product was purchased. |
| basket | This is a unique number assigned to a trip to the store. This is the trip that this product was assigned to. |
| day | This is the day that this product was purchased. The range of values is 1 to 728. |
| coupon | This is dummy variable to notify whether a coupon was used. The possible value is 1 or 0. 1 means a coupon was used. 0 means a coupon wasn’t used. |

**Store Lookup Dataset Table** In this dataset, a unique number has been assigned to the store for unique indentification, along with the zip-code where store is located.

|  |  |
| --- | --- |
| Variable | Description |
| store | This value is a unique number assigned to each store. |
| store\_zip\_code | This is the 5 digit zip code for the store. |

**Product Lookup Dataset Table** This dataset contains only 4 commodities, i.e., pasta, pasta sauce, syrup, and pancake mix over 957 different products.

|  |  |
| --- | --- |
| Variable | Description |
| upc | This is the standard 10 digit code assigned to this product. |
| product\_description | This details the product. This likely contains the name of the product. |
| commodity | This is the category of the product. The four possibly values are pasta, pasta sauce, pancake mix, or syrup. |
| brand | This is the brand name of the product. |
| product\_size | This is the size of the product. These aren’t all in the same measurement. |

**Causal Lookup Dataset Table** The below dataset describes where the product is physcially located in the store. Along with that, column feature\_desc describes the location of the advertisment in the weekly ad section.

|  |  |
| --- | --- |
| Variable | Description |
| upc | This is the standard 10 digit code assigned to this product. |
| store | This value is a unique number assigned to each store. |
| week | This notifies the week that the transaction occured. The range of values is from 1 to 104. These numbers are assigned chronologically. |
| feature\_desc | This is where the product is located on the weekly ad. |
| display\_desc | This is where the product is displayed in the store. |
| geography | This label tells you where it was purchased out of the two large regions. These two regions consist of multiple states. The value can either be 1 or 2. |

#Set the working directory#  
setwd("~/8451\_Carbo-Loading/Carbo-Loading CSV")  
#setwd("E:/UC Study Material/Spring 2020/Flex 3/BDI/8451\_Carbo-Loading/Carbo-Loading CSV")  
#Read the datasets into R#  
causal <- read\_csv("causal\_lookup.csv")  
transaction <- read\_csv("transactions.csv")  
product <-read\_csv("product\_lookup.csv")  
store <- read\_csv("store\_lookup.csv")

The below code will publish the top 5 rows of the dataset,i.e., causal lookup, transactions over 2 year period, product and store information.

#View snippets of the dataset#  
head(causal)

## # A tibble: 6 x 6  
## upc store week feature\_desc display\_desc geography  
## <chr> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 7680850108 1 68 Wrap Interior Feature Not on Display 1  
## 2 5100001212 1 66 Wrap Back Feature Not on Display 1  
## 3 5100002792 1 72 Interior Page Feature Not on Display 1  
## 4 3620000300 1 55 Wrap Interior Feature Not on Display 1  
## 5 4112907742 1 68 Wrap Interior Feature Not on Display 1  
## 6 3620000250 1 55 Wrap Interior Feature Not on Display 1

head(transaction)

## # A tibble: 6 x 11  
## upc dollar\_sales units time\_of\_transac~ geography week household store  
## <chr> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 7680~ 0.8 1 1100 2 1 125434 244  
## 2 3620~ 3.59 1 1100 2 1 125434 244  
## 3 1800~ 2.25 1 1137 2 1 108320 244  
## 4 9999~ 0.85 1 1148 2 1 162016 244  
## 5 9999~ 2.19 1 1323 2 1 89437 244  
## 6 5100~ 2.19 1 1323 2 1 89437 244  
## # ... with 3 more variables: basket <dbl>, day <dbl>, coupon <dbl>

head(product)

## # A tibble: 6 x 5  
## upc product\_description commodity brand product\_size  
## <dbl> <chr> <chr> <chr> <chr>   
## 1 111112360 VINCENT S ORIG MARINARA S pasta sauce Vincent's 25 OZ   
## 2 566300023 PINE MOUNTAIN SYRUP syrups Pine Mountain 40 OZ   
## 3 566300028 MILLER CANE SYRUP syrups Miller 19 OZ   
## 4 566300029 MILLER CANE SYRUP syrups Miller 12 OZ   
## 5 566300035 PINE MOUNTAIN SYRUP syrups Pine Mountain 19 OZ   
## 6 601011292 BARILLA MARINARA PSTA SCE pasta sauce Barilla 26 OZ

head(store)

## # A tibble: 6 x 2  
## store store\_zip\_code  
## <dbl> <dbl>  
## 1 1 37865  
## 2 2 30084  
## 3 3 30039  
## 4 4 31210  
## 5 5 30044  
## 6 6 31204

### Twitter Data Source

The final source of data we used for our projet was from [Twitter](http://apps.twitter.com). We had to create a developer account in order to get access to these tweets. In this project, we were hoping to pull this data manually to get customer opinions about brands. The way we pulled these tweets down manually was through using the search tweets package in R, and we searched through twitter to find these tweets by searching on the brand name and the food category of the brand. Next, we cleaned these results by focusing just on the tweet, removing emojis, removing urls, lowering the case of the tweets, removing the punctation, removing the numbers, removing stop words, removing white space, and removing additional words that we don’t want to evaluate. After we cleaned the tweets, we separated the tweets into words, counted these words, and filtered on the top 10 words. After collecting these results, we created a data frame from all these results. If a brand and category didn’t show a result, we put NA for these word results. Below is a table of the dataset we created manually, and this table shows the variables and description.

#The below is an example of the code we wrote for each brand in order to get this twitter data. We had to alternate the brand name and commodity for each run of the twitter data#  
#Barilla Tweets#  
barilla <- c("barilla", "pasta")  
barilla\_search <- paste(barilla, collapse = " AND ")  
barilla\_tweets <- search\_tweets(q = barilla\_search, n = 1000, lang = "en", include\_rts = FALSE)  
tibble(barilla\_tweets)  
  
head(barilla\_tweets$text)  
  
barilla\_text <- barilla\_tweets$text  
  
corpus\_barilla <- Corpus(VectorSource(barilla\_tweets$text))  
  
barilla\_emoji\_clean <- tm\_map(corpus\_barilla, content\_transformer(gsub), pattern = "\\W", replace = " ")  
  
remove\_urls <- function(x) gsub("http[^[:space:]]\*", "", x)  
  
url\_clean\_barilla<- tm\_map(barilla\_emoji\_clean, content\_transformer(remove\_urls))  
  
lower\_barilla <- tm\_map(url\_clean\_barilla, content\_transformer(tolower))  
  
remove\_items <- function(x) gsub("[^[:alpha:][:space:]]\*", "", x)  
remove\_barilla <- tm\_map(lower\_barilla, content\_transformer(remove\_items))  
  
remove\_punct\_barilla <- tm\_map(remove\_barilla, removePunctuation)  
  
remove\_num\_barilla <- tm\_map(remove\_punct\_barilla, removeNumbers)  
  
remove\_stop\_barilla <- tm\_map(remove\_num\_barilla, removeWords, stopwords("english"))  
  
remove\_white\_barilla <- tm\_map(remove\_stop\_barilla, stripWhitespace)  
  
remove\_words\_barilla <- tm\_map(remove\_white\_barilla, removeWords, c("barilla", "barillas", "cbarilla", "amp", "pasta", "pastas"))  
  
matrix\_barilla <- TermDocumentMatrix((remove\_words\_barilla))  
matrix\_barilla\_2 <- as.matrix(matrix\_barilla)  
sort\_barilla <- sort(rowSums(matrix\_barilla\_2), decreasing = TRUE)  
data\_frame\_barilla <- data.frame(word = names(sort\_barilla), freq = sort\_barilla)  
  
top\_10\_barilla <- data\_frame\_barilla %>%  
 arrange(desc(freq)) %>% head(10)  
  
barilla\_final <- crossing(top\_10\_barilla, brand = "Barilla")  
barilla\_final  
  
#After we found this data, we combined the datasets together that had tweets that showed up. The code we used is below#  
twitter\_data <- rbind(miller\_final, barilla\_final, golden\_eagle\_final, rr\_final, alaga\_final, creamette\_final, bisquick\_final, hungry\_jack\_final, kraft\_final, dececco\_final, eden\_final, pomi\_final, hunts\_final, vita\_final, mothers\_final, mueller\_final, la\_moderna\_final, aunt\_jemima\_final, tree\_of\_life\_final, ronzoni\_final, san\_giorgio\_final, ragu\_final, bertolli\_final, davinci\_final, kellogg\_final, colavita\_final, san\_marzano\_final, classico\_final, krusteaz\_final, pioneer\_final, buitoni\_final, mrs\_butterworth\_final, raos\_final, karo\_final, prego\_final, joeys\_final, smuckers\_final, daves\_final, brothers\_final, orzo\_final, howards\_final, knotts\_final, hodgson\_mills\_final, amore\_final, no\_yolks\_final, rf\_final, hse\_final, maple\_grove\_final, lyles\_final, eddie\_final, northwoods\_final, al\_dente\_final, cucina\_final, silver\_palate\_final, quinoa\_final, m\_c\_final, rac\_final, moms\_final, sinatras\_final)  
  
#After we finished combining the datasets, we saved the dataset as a csv file. The code is below for how we saved this dataset#  
write.csv(twitter\_data, "twitter\_data.csv")

#This is the code we used to read the dataset into R#  
#Set the working directory#  
setwd("~/8451\_Carbo-Loading/Carbo-Loading CSV")  
#Read the dataset into R#  
twitter\_data <- read\_csv("twitter\_data.csv")

## Parsed with column specification:  
## cols(  
## word = col\_character(),  
## freq = col\_double(),  
## brand = col\_character()  
## )

#See a preview of the dataset#  
head(twitter\_data)

## # A tibble: 6 x 3  
## word freq brand   
## <chr> <dbl> <chr>   
## 1 brown 1 Miller  
## 2 butter 1 Miller  
## 3 cane 1 Miller  
## 4 cornbread 1 Miller  
## 5 maple 1 Miller  
## 6 pure 1 Miller

**Twitter Dataset Table**

|  |  |
| --- | --- |
| Variable | Description |
| word | These are the words that showed up in the top 10 filter. |
| freq | These are the counts of the word showing up in the tweets. |
| brand | This is the name of the brand. |

### Zip Data Source

**Zip Dataset Table**

|  |  |
| --- | --- |
| Variable | Description |
| zip | zip code |

#Set the working directory#  
setwd("~/8451\_Carbo-Loading/Carbo-Loading CSV")  
#setwd("E:/UC Study Material/Spring 2020/Flex 3/BDI/8451\_Carbo-Loading/Carbo-Loading CSV")  
  
#Read the zip dataset into R#  
zip <- read\_csv("zip.csv")

#See a partial view of the dataset#  
head(zip)

## # A tibble: 6 x 8  
## Zip City State Latitude Longitude Timezone `Daylight savings ti~ geopoint  
## <dbl> <chr> <chr> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 71937 Cove AR 34.4 -94.4 -6 1 34.4  
## 2 72044 Edgemo~ AR 35.6 -92.2 -6 1 35.6  
## 3 56171 Sherbu~ MN 43.7 -94.7 -6 1 43.7  
## 4 49430 Lamont MI 43.0 -85.9 -5 1 43.0  
## 5 52585 Richla~ IA 41.2 -92.0 -6 1 41.2  
## 6 47520 Cannel~ IN 37.9 -86.7 -5 0 37.9

### Data Integration Process

The team has integrated the data based on the questions that we need to address:

1. To find the impact of product placement and ads on the sales of a product in a particular store, we are joining causal and transaction tables with the store, upc, and week as the common attribute. We are analyzing this combined data for approximately a year to negate any seasonal bias.
2. Since our dataset is huge and consists of many columns that do not influence the analysis, we are dropping them. They just eat up memory space and slow down processing. So we are using the columns only which add value to our analysis viz, upc, units, week, store, day and product description.

#Transformations of the datasets#  
#Clean the files before the inner join for Question 2#  
causal\_select <- causal %>%  
 select(upc, store, week, feature\_desc, display\_desc)  
transaction\_clean <- transaction %>%  
 filter(week > 42) %>%  
 select(upc, dollar\_sales, units, week, store)  
  
#Make some transformations of the data for Question 3#  
product\_transform <- transform(product, upc = as.character(upc))  
store\_transform <- transform(store, store\_zip\_code = as.character(store\_zip\_code))  
names(store\_transform)[2] <- "Zip"  
zip\_transform <- transform(zip, Zip = as.character(Zip))  
  
#Select the columns needed from each of the datasets for Question 3#  
transaction\_select <- transaction %>%  
 select(upc, units, week, store, day)  
product\_select <- product\_transform %>%  
 select(upc, product\_description)  
store\_select <- store\_transform %>%  
 select(store, Zip)

#First Inner Join for Question 1#  
prod\_trans <- inner\_join(transaction, product\_transform, by = c("upc"))  
head(prod\_trans)

## # A tibble: 6 x 15  
## upc dollar\_sales units time\_of\_transac~ geography week household store  
## <chr> <dbl> <dbl> <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 7680~ 0.8 1 1100 2 1 125434 244  
## 2 3620~ 3.59 1 1100 2 1 125434 244  
## 3 1800~ 2.25 1 1137 2 1 108320 244  
## 4 9999~ 0.85 1 1148 2 1 162016 244  
## 5 9999~ 2.19 1 1323 2 1 89437 244  
## 6 5100~ 2.19 1 1323 2 1 89437 244  
## # ... with 7 more variables: basket <dbl>, day <dbl>, coupon <dbl>,  
## # product\_description <chr>, commodity <chr>, brand <chr>, product\_size <chr>

#Group by brand to show the sum of sales for Question 1#  
prod\_trans\_group <- prod\_trans %>%  
 group\_by(brand, commodity) %>%  
 summarise(brand\_total\_sales = sum(dollar\_sales, na.rm = TRUE)) %>%  
 arrange(brand\_total\_sales)

#Second Inner Join for Question 1#  
#Left join on group data and twitter data#  
prod\_trans\_group\_twitter <- left\_join(prod\_trans\_group, twitter\_data, by = "brand")  
head(prod\_trans\_group\_twitter)

## # A tibble: 6 x 5  
## # Groups: brand [2]  
## brand commodity brand\_total\_sales word freq  
## <chr> <chr> <dbl> <chr> <dbl>  
## 1 La Russa pasta 1.07 <NA> NA  
## 2 Quinoa pasta 1.94 beans 14  
## 3 Quinoa pasta 1.94 bread 12  
## 4 Quinoa pasta 1.94 chicken 15  
## 5 Quinoa pasta 1.94 eat 12  
## 6 Quinoa pasta 1.94 free 12

#Inner join for Question 2#  
causal\_trans <- inner\_join(causal\_select, transaction\_clean, by = c("upc", "store", "week"))  
tibble(causal\_trans)

## # A tibble: 381,746 x 1  
## causal\_trans$upc $store $week $feature\_desc $display\_desc $dollar\_sales  
## <chr> <dbl> <dbl> <chr> <chr> <dbl>  
## 1 3620000300 1 55 Wrap Interio~ Not on Displ~ 1.5  
## 2 4112907742 1 68 Wrap Interio~ Not on Displ~ 2   
## 3 4420979129 1 66 Interior Pag~ Not on Displ~ 2.5  
## 4 3620001375 1 56 Interior Pag~ Not on Displ~ 1.5  
## 5 4112907700 1 68 Wrap Interio~ Not on Displ~ 2   
## 6 3620001368 1 56 Interior Pag~ Not on Displ~ 3   
## 7 7680851613 1 67 Wrap Interio~ Not on Displ~ 0.8  
## 8 7680851613 1 67 Wrap Interio~ Not on Displ~ 1.6  
## 9 3620000488 1 56 Interior Pag~ Not on Displ~ 1.5  
## 10 3620000482 1 56 Interior Pag~ Not on Displ~ 1.5  
## # ... with 381,736 more rows, and 1 more variable: $units <dbl>

#Inner join on all these datasets for Question 3#  
transaction\_product <- inner\_join(transaction\_select, product\_select, by = "upc")  
head(transaction\_product)

## # A tibble: 6 x 6  
## upc units week store day product\_description   
## <chr> <dbl> <dbl> <dbl> <dbl> <chr>   
## 1 7680850106 1 1 244 1 BARILLA ANGEL HAIR   
## 2 3620000470 1 1 244 1 BERTOLLI TOM&BASIL SAUCE   
## 3 1800028064 1 1 244 1 H J PANCK BTRMLK COMP MIX   
## 4 9999985067 1 1 244 1 PRIVATE LABEL VERMICELLI   
## 5 9999985131 1 1 244 1 PRIVATE LABEL IMPORTED LASAGNA  
## 6 5100002794 1 1 244 1 PREGO SPAG SAUCE MEAT

transaction\_product\_store <- inner\_join(transaction\_product, store\_select, by = "store")  
head(transaction\_product\_store)

## # A tibble: 6 x 7  
## upc units week store day product\_description Zip   
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr>  
## 1 7680850106 1 1 244 1 BARILLA ANGEL HAIR 40222  
## 2 3620000470 1 1 244 1 BERTOLLI TOM&BASIL SAUCE 40222  
## 3 1800028064 1 1 244 1 H J PANCK BTRMLK COMP MIX 40222  
## 4 9999985067 1 1 244 1 PRIVATE LABEL VERMICELLI 40222  
## 5 9999985131 1 1 244 1 PRIVATE LABEL IMPORTED LASAGNA 40222  
## 6 5100002794 1 1 244 1 PREGO SPAG SAUCE MEAT 40222

#Additional inner join for question 3 for creating the zip code map#  
#Group the data by zip code and the sum the quanity of products sold in these zip codes#  
head(transaction\_product\_store)

## # A tibble: 6 x 7  
## upc units week store day product\_description Zip   
## <chr> <dbl> <dbl> <dbl> <dbl> <chr> <chr>  
## 1 7680850106 1 1 244 1 BARILLA ANGEL HAIR 40222  
## 2 3620000470 1 1 244 1 BERTOLLI TOM&BASIL SAUCE 40222  
## 3 1800028064 1 1 244 1 H J PANCK BTRMLK COMP MIX 40222  
## 4 9999985067 1 1 244 1 PRIVATE LABEL VERMICELLI 40222  
## 5 9999985131 1 1 244 1 PRIVATE LABEL IMPORTED LASAGNA 40222  
## 6 5100002794 1 1 244 1 PREGO SPAG SAUCE MEAT 40222

zip\_map\_data\_2 <- transaction\_product\_store %>%  
 group\_by(Zip) %>%  
 summarise(Total\_Quantity\_Zip = sum(units, na.rm = TRUE))  
  
#Inner join on zip and transaction product store#  
transaction\_product\_store\_zip <- inner\_join(zip\_map\_data\_2, zip\_transform, by = "Zip")  
head(transaction\_product\_store\_zip)

## # A tibble: 6 x 9  
## Zip Total\_Quantity\_~ City State Latitude Longitude Timezone  
## <chr> <dbl> <chr> <chr> <dbl> <dbl> <dbl>  
## 1 29063 14713 Irmo SC 34.1 -81.2 -5  
## 2 29204 7391 Colu~ SC 34.0 -81.0 -5  
## 3 29205 8409 Colu~ SC 34.0 -81.0 -5  
## 4 29210 9312 Colu~ SC 34.0 -81.1 -5  
## 5 29229 17538 Colu~ SC 34.1 -80.9 -5  
## 6 29572 7961 Myrt~ SC 33.8 -78.8 -5  
## # ... with 2 more variables: Daylight.savings.time.flag <dbl>, geopoint <dbl>

#### Schema Alignment

At the initial stages of your big data analysis, you’re not going to possess the identical level of control over data definitions as you are doing together with your operational data. However, once you have got identified the patterns that are most relevant to your business, you wish the potential to map data elements to a standard definition. That common definition is then carried forward into operational data, data warehouses, reporting, and business processes.

The traditional approach of schema alignment consists of three main steps - creating a mediated schema, attribute matching and schema mapping. First, a mediated schema is created by unifying the schemas of the data sources being integrated. Queries to the data integration system are formulated on this mediated schema. Next is the attribute matching step which specifies how attribute values from different schemas are to be matched for equality in the mediated schema. In many cases the attribute correspondence is one-to-one however, sometimes one attribute may correspond to a combination of several attributes in the source schema. The last stage is the Schema Mapping which is built between each source schema and the mediated schema. The problem of establishing associations between data under different schemas is at the core of many data integration and data sharing tasks.

#### Record Linkage

After creating a mediated schema by integrating different sources, we observe that different sources provide value for the same attribute of the same entity. These values may often differ due to mistyping, multiple naming conventions and so on. For instance, the column for date of birth can be named as ‘birth date’ in one schema and ‘dob’ in the other. These representational differences make it hard to link such similar records even though they refer to the same entity. Record linkage attempts to link such millions of records obtained from tens to thousands of data sources.

The process of record linkage starts with pairwise matching where a pair of records are compared to make a local decision of whether they refer to the same entity or not. But often this local decision making may not be globally consistent.

#### Data Fusion

Often while combining data from different sources to create a mediated data schema, we find that such different sources provide conflicting values for the same attribute of the same entity. These conflicts can arise because of mistyping, miscalculations or out of date information which has not been updated consistently across all the databases. For instance, let us consider that the takeoff time for a flight has changed starting January 2020 but this information has not been updated across all the databases which contains information about this flight. This leads to conflicting information which can be confusing and also under certain circumstances, harmful. Data fusion attempts at combining such records that refer to the same real-world entity into a single representation by resolving possible conflicts from different data sources.

###Conclusion for Data Integration in Carbon-Loading datasets For the datasets we have used, we have not encountered much issues with schema alignment, record linkage and record linkage. This is primarily since the source of our datasets is uniform. We have received all the four datasets used in our analysis from 8451. So, our datasets were clean and homogeneous. Only our zip dataset was in csv format while the other three were in excel. This would have required some work on schema alignment had we not used a powerful analysis tool like R which can read and analysis data in different formats like excel and csv without any additional efforts being made.

### Final Results and Analyzing These Results

#Question 1 Editing and Results#  
#Top 5 worst performing brands overall#  
head(prod\_trans\_group, 5)

## # A tibble: 5 x 3  
## # Groups: brand [5]  
## brand commodity brand\_total\_sales  
## <chr> <chr> <dbl>  
## 1 La Russa pasta 1.07  
## 2 Quinoa pasta 1.94  
## 3 Antoine's pasta 2.19  
## 4 Bionature pasta 2.19  
## 5 Pastariso pasta 2.95

#Top 5 Worst performing brands for pasta#  
prod\_trans\_group\_pasta <- prod\_trans\_group %>%  
 filter(commodity == "pasta") %>%  
 arrange(brand\_total\_sales)  
head(prod\_trans\_group\_pasta, 5)

## # A tibble: 5 x 3  
## # Groups: brand [5]  
## brand commodity brand\_total\_sales  
## <chr> <chr> <dbl>  
## 1 La Russa pasta 1.07  
## 2 Quinoa pasta 1.94  
## 3 Antoine's pasta 2.19  
## 4 Bionature pasta 2.19  
## 5 Pastariso pasta 2.95

#Top 5 Worst performing brands for pasta sauce#  
prod\_trans\_group\_pasta\_sauce <- prod\_trans\_group %>%  
 filter(commodity == "pasta sauce") %>%  
 arrange(brand\_total\_sales)  
head(prod\_trans\_group\_pasta\_sauce, 5)

## # A tibble: 5 x 3  
## # Groups: brand [5]  
## brand commodity brand\_total\_sales  
## <chr> <chr> <dbl>  
## 1 M C pasta sauce 5.29  
## 2 Pomi pasta sauce 5.58  
## 3 B F pasta sauce 5.79  
## 4 Ferrara pasta sauce 5.97  
## 5 RR pasta sauce 5.99

#Top 5 Worst performing brands for syrups#  
prod\_trans\_group\_syrups <- prod\_trans\_group %>%  
 filter(commodity == "syrups") %>%  
 arrange(brand\_total\_sales)  
head(prod\_trans\_group\_syrups, 5)

## # A tibble: 5 x 3  
## # Groups: brand [5]  
## brand commodity brand\_total\_sales  
## <chr> <chr> <dbl>  
## 1 Braswell syrups 2.99  
## 2 DaVinci syrups 7.18  
## 3 Knott's syrups 8.07  
## 4 Lyles syrups 10.5   
## 5 Vermont Gold syrups 14.0

#Top 5 Worst performing brands for pancake mixes#  
prod\_trans\_group\_pancake\_mixes <- prod\_trans\_group %>%  
 filter(commodity == "pancake mixes") %>%  
 arrange(brand\_total\_sales)  
head(prod\_trans\_group\_pancake\_mixes, 5)

## # A tibble: 5 x 3  
## # Groups: brand [5]  
## brand commodity brand\_total\_sales  
## <chr> <chr> <dbl>  
## 1 Lund Swede pancake mixes 8.76  
## 2 Fastshake pancake mixes 39.6   
## 3 Osem Bissli pancake mixes 106.   
## 4 M W Flapstax pancake mixes 2084.   
## 5 Classique pancake mixes 2675.

#We couldn't show the shiny ggplot word cloud because it is interactive and it doesn't show up in word documents. You can see the world cloud in the presentation. We will be providing images of the cloud to show you what it looks like#

#Editing of the Combined Dataset to Find the Answers for Question 2#  
#Feature Description for Sales#  
causal\_trans\_feature\_dollar <- causal\_trans %>%  
 select(feature\_desc, dollar\_sales)%>%  
 group\_by(feature\_desc) %>%  
 summarise(Total\_Dollar\_Sales = sum(dollar\_sales, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Dollar\_Sales))  
causal\_trans\_feature\_dollar

## # A tibble: 8 x 2  
## feature\_desc Total\_Dollar\_Sales  
## <chr> <dbl>  
## 1 Interior Page Feature 240347.  
## 2 Not on Feature 172678.  
## 3 Front Page Feature 105429.  
## 4 Wrap Interior Feature 22507.  
## 5 Back Page Feature 17707.  
## 6 Wrap Front Feature 11098.  
## 7 Wrap Back Feature 9492.  
## 8 Interior Page Line Item 9378.

#Feature Description for Units#  
causal\_trans\_feature\_units <- causal\_trans %>%  
 select(feature\_desc, units)%>%  
 group\_by(feature\_desc) %>%  
 summarise(Total\_Units = sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Units))  
causal\_trans\_feature\_units

## # A tibble: 8 x 2  
## feature\_desc Total\_Units  
## <chr> <dbl>  
## 1 Interior Page Feature 196841  
## 2 Not on Feature 136526  
## 3 Front Page Feature 106489  
## 4 Wrap Interior Feature 19338  
## 5 Back Page Feature 17476  
## 6 Wrap Front Feature 11210  
## 7 Wrap Back Feature 7467  
## 8 Interior Page Line Item 5965

#Feature Description Sales divided by Units (aka average price)#  
causal\_trans\_feature\_units\_dollar <- causal\_trans %>%  
 select(feature\_desc, units, dollar\_sales)%>%  
 group\_by(feature\_desc) %>%  
 summarise(Sales\_Divided\_By\_Units = sum(dollar\_sales, na.rm = TRUE)/sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Sales\_Divided\_By\_Units))  
causal\_trans\_feature\_units\_dollar

## # A tibble: 8 x 2  
## feature\_desc Sales\_Divided\_By\_Units  
## <chr> <dbl>  
## 1 Interior Page Line Item 1.57   
## 2 Wrap Back Feature 1.27   
## 3 Not on Feature 1.26   
## 4 Interior Page Feature 1.22   
## 5 Wrap Interior Feature 1.16   
## 6 Back Page Feature 1.01   
## 7 Front Page Feature 0.990  
## 8 Wrap Front Feature 0.990

#Display\_desc sales#  
causal\_trans\_display\_dollar <- causal\_trans %>%  
 select(display\_desc, dollar\_sales)%>%  
 group\_by(display\_desc) %>%  
 summarise(Total\_Dollar\_Sales = sum(dollar\_sales, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Dollar\_Sales))  
causal\_trans\_display\_dollar

## # A tibble: 11 x 2  
## display\_desc Total\_Dollar\_Sales  
## <chr> <dbl>  
## 1 Not on Display 324395.  
## 2 Rear End Cap 79471.  
## 3 In-Shelf 63270.  
## 4 Front End Cap 31889.  
## 5 Secondary Location Display 23581.  
## 6 In-Aisle 21124.  
## 7 Promo/Seasonal Aisle 13619.  
## 8 Mid-Aisle End Cap 13288.  
## 9 Store Rear 10016.  
## 10 Store Front 6147.  
## 11 Side-Aisle End Cap 1835.

#Display\_desc units sold#  
causal\_trans\_display\_units <- causal\_trans %>%  
 select(display\_desc, units)%>%  
 group\_by(display\_desc) %>%  
 summarise(Total\_Units = sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Total\_Units))  
causal\_trans\_display\_units

## # A tibble: 11 x 2  
## display\_desc Total\_Units  
## <chr> <dbl>  
## 1 Not on Display 281389  
## 2 Rear End Cap 68814  
## 3 In-Shelf 52084  
## 4 Front End Cap 28086  
## 5 Secondary Location Display 16422  
## 6 In-Aisle 16219  
## 7 Promo/Seasonal Aisle 12359  
## 8 Mid-Aisle End Cap 11084  
## 9 Store Rear 8349  
## 10 Store Front 5256  
## 11 Side-Aisle End Cap 1250

#Display\_desc sales divided by units (aka average price)#  
causal\_trans\_display\_units\_dollar <- causal\_trans %>%  
 select(display\_desc, units, dollar\_sales)%>%  
 group\_by(display\_desc) %>%  
 summarise(Sales\_Divided\_By\_Units = sum(dollar\_sales, na.rm = TRUE)/sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Sales\_Divided\_By\_Units))  
causal\_trans\_display\_units\_dollar

## # A tibble: 11 x 2  
## display\_desc Sales\_Divided\_By\_Units  
## <chr> <dbl>  
## 1 Side-Aisle End Cap 1.47  
## 2 Secondary Location Display 1.44  
## 3 In-Aisle 1.30  
## 4 In-Shelf 1.21  
## 5 Store Rear 1.20  
## 6 Mid-Aisle End Cap 1.20  
## 7 Store Front 1.17  
## 8 Rear End Cap 1.15  
## 9 Not on Display 1.15  
## 10 Front End Cap 1.14  
## 11 Promo/Seasonal Aisle 1.10

#Code to get Answers for Question 3#  
#Part 1#  
#Group the data by product and find the top 5 products sold by quantity#  
product\_total\_top\_5 <-transaction\_product\_store %>%  
 group\_by(product\_description) %>%  
 summarise(Sum\_Quantity = sum(units, na.rm = TRUE)) %>%  
 arrange(desc(Sum\_Quantity)) %>%  
 head(5)  
product\_total\_top\_5

## # A tibble: 5 x 2  
## product\_description Sum\_Quantity  
## <chr> <dbl>  
## 1 PRIVATE LABEL THIN SPAGHETTI 228331  
## 2 PRIVATE LABEL SPAGHETTI REGULAR 215816  
## 3 PRIVATE LABEL ELBOW MACARONI 118911  
## 4 PRIVATE LABEL ANGEL HAIR PASTA 111764  
## 5 RAGU TRADITIONAL PLAIN 100617

#Part 2#  
#Group the data by day and product description and create a calculation to sum the units by day#  
combine\_day\_sum <- transaction\_product\_store %>%  
 group\_by(day, product\_description) %>%  
 summarise(Sum\_Quantity\_by\_Day = sum(units, na.rm = TRUE))  
#Preview of dataset for paper#  
head(combine\_day\_sum)

## # A tibble: 6 x 3  
## # Groups: day [1]  
## day product\_description Sum\_Quantity\_by\_Day  
## <dbl> <chr> <dbl>  
## 1 1 (S)BARILLA GEMELLI PASTA 13  
## 2 1 (S)BARILLA RIGATONI PASTA 22  
## 3 1 (S)PREGO RICOTTA PARMESAN 4  
## 4 1 (S)RAGU CHNKY GRDNSTYL GR 18  
## 5 1 \*HODGE WHEAT SPAGHETTI 25  
## 6 1 A J BTRMLK COMP PNCK MIX 17

#The shiny table we created to show these results can be found on website for the presentation. The image below is a snippet of it. The code we used to create this table is found in the additional code section of the paper.#

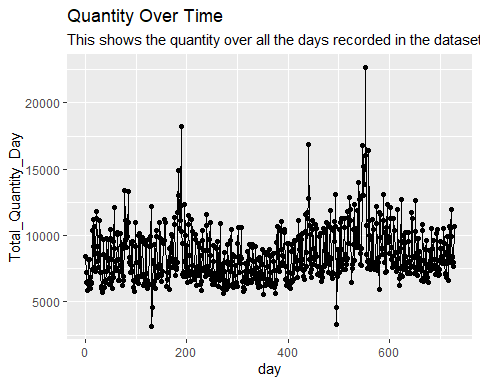
#Part 3#  
#Group the data by day, sum the units of the day, and arrange it by day#  
combine\_day\_time <- transaction\_product\_store %>%  
 group\_by(day) %>%  
 summarise(Total\_Quantity\_Day = sum(units, na.rm = TRUE)) %>%  
 arrange(day)  
#Preview the dataset#  
head(combine\_day\_time)

## # A tibble: 6 x 2  
## day Total\_Quantity\_Day  
## <dbl> <dbl>  
## 1 1 8390  
## 2 2 7226  
## 3 3 6438  
## 4 4 6540  
## 5 5 5845  
## 6 6 6348

#Find the top 5 days when products were sold#  
top\_5\_days <- combine\_day\_time %>%  
 arrange(desc(Total\_Quantity\_Day)) %>%  
 head(5)  
top\_5\_days

## # A tibble: 5 x 2  
## day Total\_Quantity\_Day  
## <dbl> <dbl>  
## 1 553 22647  
## 2 189 18203  
## 3 441 16896  
## 4 547 16791  
## 5 559 16382

#Create a line graph showing the quantity sold over time#  
ggplot(data = combine\_day\_time, aes(x = day, y = Total\_Quantity\_Day, group=1)) +  
 geom\_line() +  
 geom\_point() +  
 ggtitle("Quantity Over Time", subtitle = "This shows the quantity over all the days recorded in the dataset.")



#Part 4#  
#Select the columns we need for the zip data and arrange it in descending order by Total Quanity Sold#  
zip\_map\_data <- transaction\_product\_store\_zip %>%  
 select(Zip, Latitude, Longitude, Total\_Quantity\_Zip) %>%  
 arrange(desc(Total\_Quantity\_Zip))  
#This is a preview of the dataset we used to create the map in our presentation. Also, it shows the Top 5 zip codes for quanity sold#  
head(zip\_map\_data, 5)

## # A tibble: 5 x 4  
## Zip Latitude Longitude Total\_Quantity\_Zip  
## <chr> <dbl> <dbl> <dbl>  
## 1 37211 36.1 -86.7 76796  
## 2 47150 38.3 -85.8 69875  
## 3 40502 38.0 -84.5 64490  
## 4 30024 34.1 -84.1 54880  
## 5 30064 33.9 -84.6 51702

#We can't display the map because it is interactive. Below is an image of what the maap looks like. The code we used will be found in the additional code section of the paper#

###Conclusion

From our study, we find that the average sale of the products we have considered across different stores in the city have been fairly consistent. There have been no considerable spikes except a few non-seasonal ones which we can consider as outliers since there is no specific trend to it. We feel that this is a fair justification since we are dealing with basic commodity products here. The sale of pasta and pancake mix can not be tied to any particular day or season logically. Their sale has been consistently high as can be seen from our analysis across all the stores in the city so Kroger should always keep an ample amount of stock always in their inventory, more for one particular brand compared to some other as can be seen from our analysis.

We may want to change the products we are analyzing on for our future study to observe if there has been any pronounced trend in their sale. For instance, we may want to consider the sale of ice cream which should ideally sale more in summer than in winter.

### Future Steps

In the future, we intend to study if there is any interaction between related products like pasta and pasta sauce and syrup and pancake mix and determine if there is actually any association between the sale of these products. We shall also study if placing similar products together actually increases the sale of either of the products. For instance, we expect that putting pasta and pasta sauce together will logically result in increased sale for both the products rather than putting pasta and syrup together.

We also want to do an association analysis between brands for such related products like do people who buy Pasta Sauce of Brand X also buys Pasta of Brand Y along with it. In such cases, we will be able to come up with better product placement and ad ideas.

### Bibliography

“Exporting a dataset from R” <http://www.instantr.com/2012/12/11/exporting-a-dataset-from-r/> “Scatter Plots on Maps in R” <https://plot.ly/r/scatter-plots-on-maps/> “US Zip Code Latitude and Longitude” <https://public.opendatasoft.com/explore/dataset/us-zip-code-latitude-and-longitude/export/> “R - Add a new column to the data frame and duplicate the existing rows” <https://stackoverflow.com/questions/48217301/r-add-a-new-column-to-the-data-frame-and-duplicate-the-existing-rows> “Text Mining Twitter Data With TidyText in R” <https://www.earthdatascience.org/courses/earth-analytics/get-data-using-apis/text-mining-twitter-data-intro-r/> “Text mining: Twitter extraction and stepwise guide to generate a word cloud” <https://towardsdatascience.com/text-mining-twitter-extraction-and-stepwise-guide-to-generate-a-word-cloud-a2c9d626008d> “Carbo-Loading: A Relational Database” <https://www.8451.com/area51> “Search Tweets” <https://developer.twitter.com/en/docs>

### Additonal Significant Code