INSIGHTS INTO AN 84.51 KROGER DATASET

Zhe Huang
Krti Tallam
Siddharth Ahuja
Jiewen Wu

STAT 443

Fall 2018, Group 6

TABLE OF CONTENTS

INSIGHTS INTO AN 84.51 KROGER DATASET

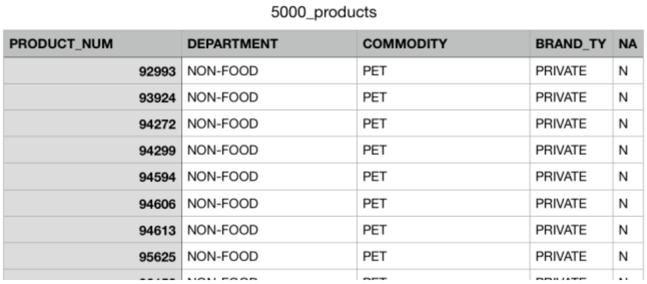
DECEMBER 12TH, 2018
STAT 443

INTRODUCTION

Over the course of this Fall 2018 semester, Group 6 of the Statistics 443 class has been working with 84.51, a subsidiary of Kroger, to study the possibilities of increasing future sales within the company. 84.51 utilizes data to provide customer insights through the use of a proprietary suite of tools and technologies, in order to uncover relevant customer patterns. Thus far, the company has analyzed more than 60 million house-holds worth of data, delivered 1 billion personalized offers last year, has obtained more than 10 petabytes worth of customer data, has analyzed approximately 3 billion shopping baskets, has 1,250 CPG partners, and has accumulated 138 data-scoring models. Our fundamental question with this project was the following: 1) Based on the information from our data analyses, what are the current sales trends, and how can Kroger increase future sales? This included demographic and geographic factors, product performance, time series models, and looking at outliers and anomalies. 2) What are the limitations of the data we currently have available to us? The scale of this increase, as well as most details of the project, were left for our group to break down through the process of our analysis, which we all enjoyed pursuing.

1
CUSTOMER
4
ACTIVATION
OUR APPROACH
TO DATA SCIENCE
2
DATA
3
INSIGHTS

The Dataset. Our dataset comprised of three initial datasets: Households, Products, and Transactions. Households consisted of 5,000 households, which gave us the ability to conduct in-depth demographic analysis on the dataset. Products consisted of 150,000 products across 43 categories, which allowed us to analyze the sales data across variables such as time. Finally, Transactions included more than one million transactions worth of data; in our case, all of our data was over a period of two years – from January of 2016, until December of 2017. Transactions consisted of Household information, Product information, Basket numbers, and Store Regions.



5000_products

PRODUCT_NUM	DEPARTMENT	COMMODITY	BRAND_TY	NA
92993	NON-FOOD	PET	PRIVATE	N
93924	NON-FOOD	PET	PRIVATE	N
94272	NON-FOOD	PET	PRIVATE	N
94299	NON-FOOD	PET	PRIVATE	N
94594	NON-FOOD	PET	PRIVATE	N
94606	NON-FOOD	PET	PRIVATE	N
94613	NON-FOOD	PET	PRIVATE	N
95625	NON-FOOD	PET	PRIVATE	N

METHODS

We approached this project through our knowledge and utilization of RStudio, a programming language for data analysis and visualization, and Power BI, a business analytics service of Microsoft. Initially, we looked at Spending within different categories – all four of our regions (South, West, East Central), months, weeks, age groups, marital statuses, number of households, and then within the departments and commodities. We also broke this down further, to understand the total spending by home status, defined as either a “Home Owner” or a “Renter.” We had to keep in mind that outliers and anomalies were critical in our analysis as well, given that we only had two years’ worth of data. Therefore, we decided to analyze all possible relationships that we could, and the following is an explanation of all of our analyses and results.

RESULTS

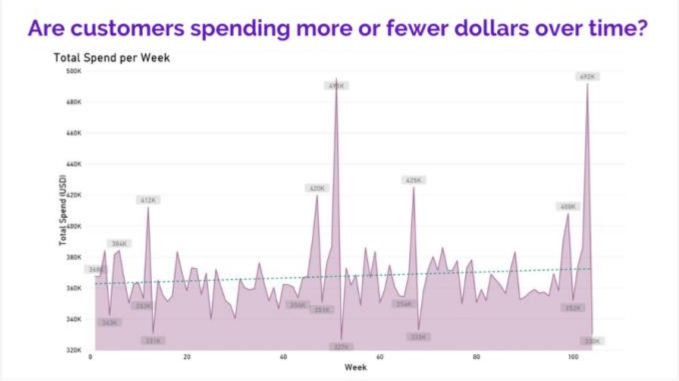


Figure 1

Total Spend per Week. The first result we looked at was if, overall, customers were spending more or less money over a given period of time. In our case, over the period of the two years, broken into weeks. We analyzed that through three types of graphs – the first one on the right, *Figure 1*, showing the slight increase in spending over the two-year period. We noticed from the other two graphs that 2,587 households spent more from year 2016 to 2017, and 2,413 households spent less between 2016 and 2017. Furthermore, when we broke this down into the average spending of baskets per week, we were able to notice some peaks in spending, as well as some anomalies in the data. For instance, we noticed peaks during Easter, as well as large peaks during the winter holiday season, in December. However, the struggle with utilizing this graph to forecast data for future years is, for instance, the December peak. Notice *Figure 2*: the peak in December of 2016 is at about 42; then, the peak in December of 2017 hits 54 in average spend per basket. Furthermore, the weeks surrounding that time frame (take two weeks before and two weeks after) are significantly different from each other – in 2016, the average spend ranges from about 36 to 42. However, in 2017, the average spend of those few weeks ranges from about 35 to 54. Therefore, it becomes a challenge to forecast this with only two-years' worth of data. Nonetheless, we utilized several other models to interpret such anomalies; we can also still suggest that one way to increase future

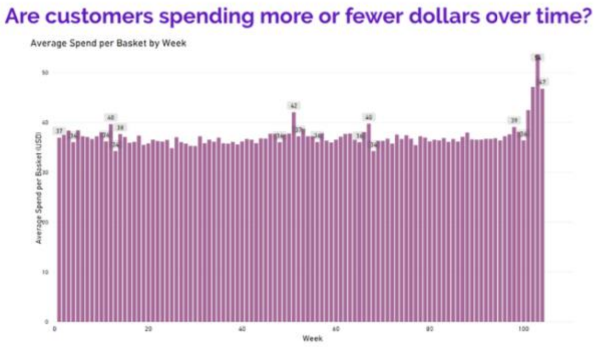


Figure 2

sales is to increase stock of products during the winter holiday season weeks. We will speak to specificities in the following passages.

Seasonality. We addressed Seasonality, analyzing the total spend per quarter, and found that Quarter 4 consistently had more money spent, between the two years, at 9.87 million dollars. Based on this knowledge, 84.51 can look into analyses of potentially increasing the stock of products during Quarter 4, for future years, as well as potentially opening up more stores, since it appears that there is a prospective for more customers and customer transactions, during this period. However, we wanted to look into what products this could be, so we did some further analysis. We will state here that all of our suggestions should be taken with thoughtfulness, as we are only estimating this based on two years' worth of data, and therefore, are likely missing out on other factors that could be playing into the dataset, despite our deep analyses.

Product analysis. We looked at the seasonality breakdown by types of products (*Figure 3*): Holiday Items, Outdoor Items, Electronics, and Floral Items, measuring trends of total spending by quarter, while plotting the number of transactions that were taking place. We found that Holiday Items and Electronics, respectively, impacted Quarter 4 spending the most; Outdoor Items impacted Quarter 2 spending the most; and Floral Items impacted Quarter 2 and then Quarter 1, respectively, the most. Therefore, we can suggest an approximate increasing of stock of Outdoor

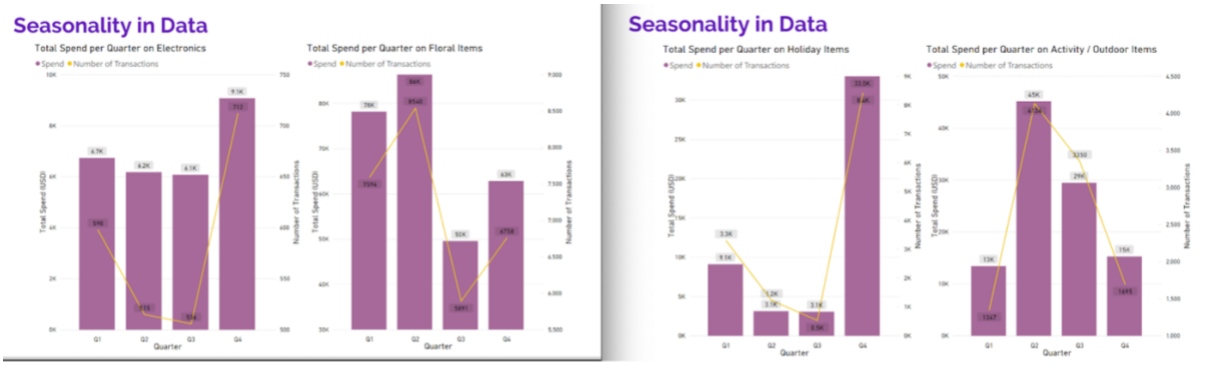


Figure 3

Items and Floral Items during Quarter 2 and an increase of Holiday Items and Electronics during Quarter 4. A drawback here was not being able to access data by store – this would have allowed us to break this case down further, to look into which stores could more specifically benefit an increase of products, and how that would affect future sales.

Geographic and Demographic factors: regions. We then analyzed geographic and demographic factors for regions, and found that the East region had both the most transactions and the most spending across the two years. We broke this down into regional spending, by week, for the two years and found that the West region actually had a greater increase in spending, while the East region only increased a little (*Figure 4A*), although in the other graph, the East region actually spent the most in these two years (*Figure 4B*). To us, this showed consistency in the East region, and we therefore could suggest to potentially increase stock of products in the East region. Furthermore, we broke this down by commodity (*Figure 4C*), and noticed that the commodity, Grocery Staples, was bought more than any other commodity, across every age group (which included ages 19 to 75 and up, and will be expanded on further, later). Therefore, based on this analysis, we can suggest a potential increase in the Grocery Staples stock, as well as maybe adding new products in that category to diversify the options and attract more customers. One

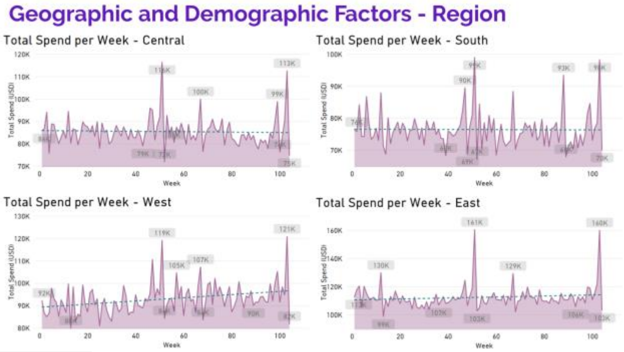


Figure 4A

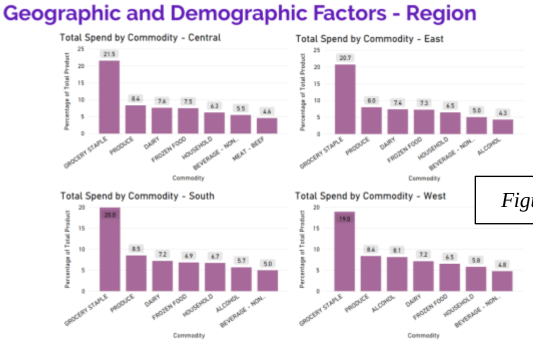


Figure 4B

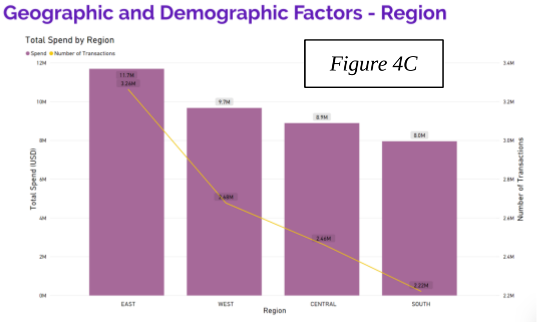


Figure 4C

interesting fact we found was alcohol replaced dairy and frozen food to become the 3rd place in the west region.

Geographic and Demographic factors: age. The next category that we analyzed was geographic and demographic factors for age groups, grouping all customers into seven age categories: 19-24, 25-34, 35-44, 45-54, 55-64, 65-74, and 75 and above. We grouped these in tens because this allowed us to analyze the data more finely than if we had done larger groups, and we did not want to make it too fine such that we were having confounding factors. We first analyzed this via bar graphs, to understand which category, overall, was spending the most (*Figure 6*). We found that the age group of 45-54 was not only spending the most in the two years, but also had the greatest number of transactions, with 9.2 million dollars spent and 2.5 million transactions made. After that, the next highest age group was 55-64 (at 7.4 million dollars spent and 2.1 million transactions made), and then the 35-44 age category (at 5.8 million dollars spent and 1.6 million transactions made). This was very interesting to us, because, at least from my experience, it does seem like my age group of around say the 19-year old's to the 34-year old's, seem to spend a lot more money on new items, such as electronics, and other items, a lot more often than older age categories do. Therefore, we decided to look at the ratio of revenue to transactions and it shows that spending was actually roughly the same across age categories. Each age group was actually spending roughly the same amount per transaction. The purpose is to

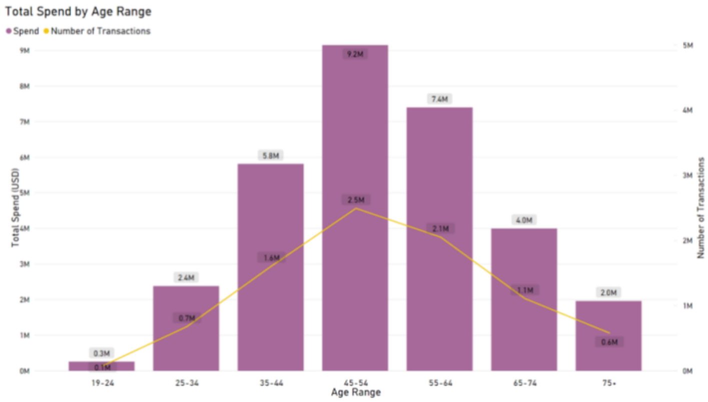


Figure 6

demonstrate that Kroger is obtaining more *customers* in the 45-54 age range, and therefore they can be targeted with more products and sales; however, there is still quite an even split of the ratio of purchases coming through, for all age groups.

We took this further and looked at how the spending was taking place over weeks, over the period of two years (*Figure 7*). We looked at weeks because, given that there were only two years' worth of data, we wanted to obtain enough data points to create a discernable trend. We noticed that between our seven graphs of analyzing each age group per week, all trends were positively increasing for all age groups, with the exception of age group 19-24. The 19-24 age group has the highest volatility (that is, when the sales were low, they were very low and vice versa) but also has the *only* decreasing trend, as the red arrow on page 6, *Figure 7* indicates). Therefore, we would suggest keeping current practices for all the other age groups and potentially looking back into the way 84.51 and Kroger were conducting practices back in 2016, as the 19-24 age group was purchasing more back in 2016. It would additionally be helpful to understand what this age group needs from a weekly basis, to better stock products for them.

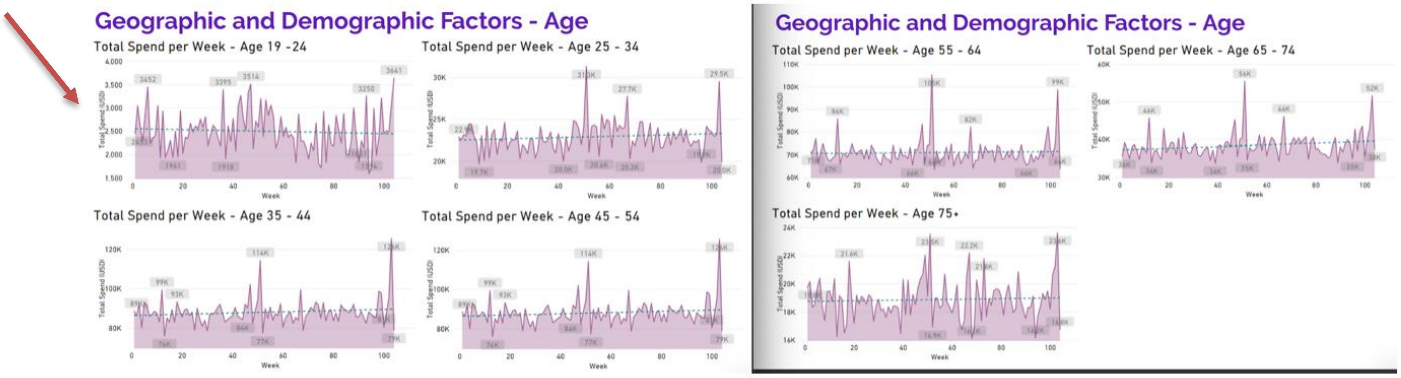


Figure 7

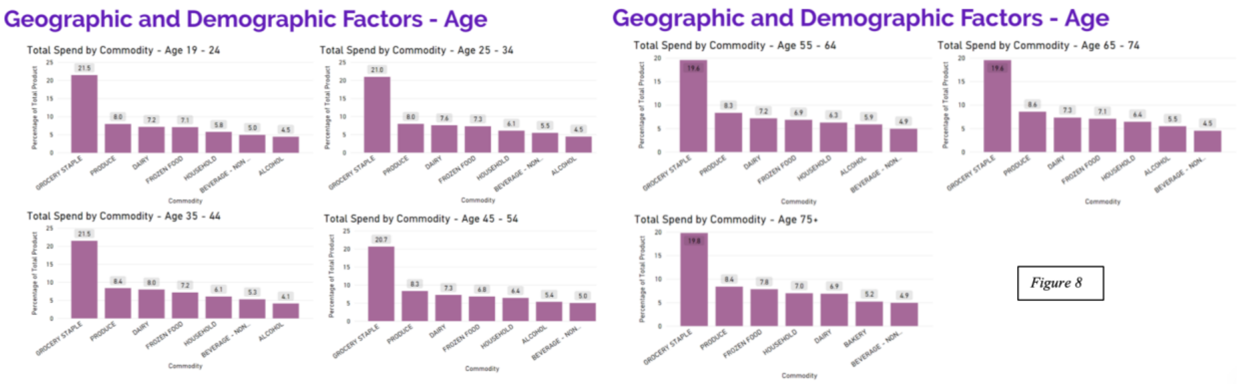


Figure 8

Continuing our analysis with age, we now add in commodity to observe what products are being purchased the most, and how knowing this could potentially increase future sales for Kroger (*Figure 8*). We noticed that the commodity sold and purchased the most across all seven age categories was the Grocery Staple commodity. Therefore, from further insight, we suggest that in order to likely increase future sales for the company, increase the stock of Grocery Staples that are available across regions. We would have been able to conduct further analysis if we had the data per store, to identify which stores may profit most from having an increase in the Grocery Staple commodity; nonetheless, the trend seems rather clear across all age groups, and therefore, we can speculate that it may be influencing profit significantly across stores.

Geographic and Demographic factors: marital and home status. After breaking this down into regions, age groups, and commodities, we decided to look at marital status, as one of our demographic factors (*Figure 9*); this proved to be significant to provide us insight into the

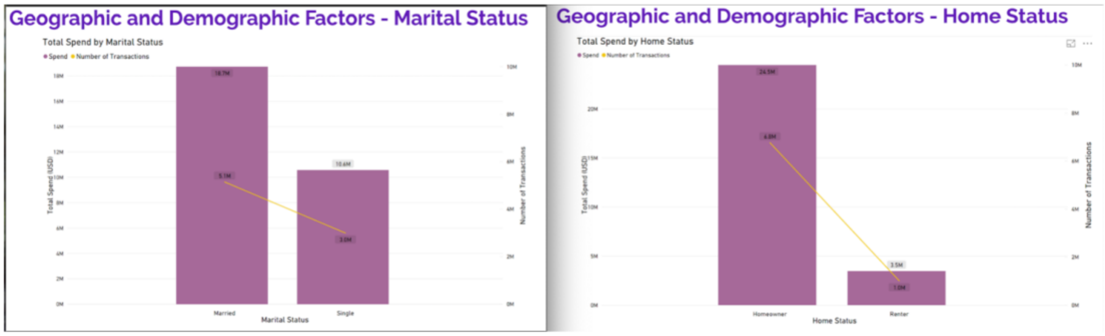


Figure 9

types of customers *not* to primarily or solely target. Total spend showed values of 18.7 million and 10.6 million dollars spent, for married and single folks respectively; and 5.1 million to 3.0 million transactions made, respectively. The total spending per week, as seen over the two years, was also significantly greater. However, the ratio of spending is once again approximately the same between Home owners and Renters – that is, each group is spending roughly the same amount per transaction. We broke this into weekly trends over the two-year period, and noticed that there were very similar peaks and dips for both groups, as well as both having positive trends in sales. However, we also noticed in our per-commodity breakdown, that customers with Single status were purchasing more Frozen Food than those with Married status, with values of 7.3 percent for Single, and 6.9 percent for Married – a significant difference. Therefore, when targeting customers with sales or competitive products (competitive being defined as products that also expand options, not necessarily reduce prices), we would suggest focusing on targeting Single customers for frozen food offers and deals and sales. Overall, however, we advise that Kroger, therefore, should not chiefly target customers based on their marital status.

Furthermore, when we look at total spending by home status (*Figure 9*), defined as either a "Home Owner" or a "Renter," we notice that the home owners spend significantly more than renters, at 25.4 million to 3.5 million dollars spent, and 6.8 million to 1.0 million transactions made, respectively. Once again however, we notice that it does appear as if Home Owners are significantly increasing sales for Kroger – however, dividing the total revenue by the transactions made, we once again notice that the ratio of spending is approximately the same. For instance, $24.5 million / 6.8 million transactions = $3.60 per transaction, and $3.5 million / 1.0 million transactions = $3.55 per transaction.

Recap and some “future increase in sales” suggestions. Having analyzed regions, age groups, commodities, marital status, and home status, we noticed that we can make a few recommendations for the company to increase future sales, from our analysis thus far: for instance, target customers in areas that have more married couples, and folks with homes, versus those who are renters. Increase product stock in those areas, and potentially increase the number of stores, and the products offered. We would suggest keeping a close eye on how that alters the current data, and adjusting as seems appropriate based on all the additional knowledge the company has. Furthermore, add more Grocery Staples across stores, as that commodity seems the most stable among almost all the factors that we analyzed.

Finally, there are critical peaks during holiday seasons, festivities, school trends: there are peaks during the second and third weeks of January, where the Spring semester for many schools begins, and students are buying lots of school supplies; therefore, a suggested increase in stock of school supplies is suggested. Then, during Easter there is another spike in the general trend; therefore, a suggestion of potentially increasing Easter product stock; then, towards the end of the year, around December, there is a huge spike for spending for the holidays, at which point we suggest also increasing stock of holiday items, for folks heading out to find that perfect gift for their families and friends, and of course, that special someone.

Geographic and Demographic factors: income level. Continuing on with geographic and demographic factors, we now took a look at the income ranges of groups; we broke it down by Less Than 35,000, 35,000 to 49,000, 50,000 to 74,000, 75,000 to 99,000, 100,000 to 150,000, and finally, More Than 150,000 dollars per year income categories, a total of six income categories (*Figure 10*). These were plotted on the x-axis against total spending per each of those categories; additionally, we added the number of transactions.

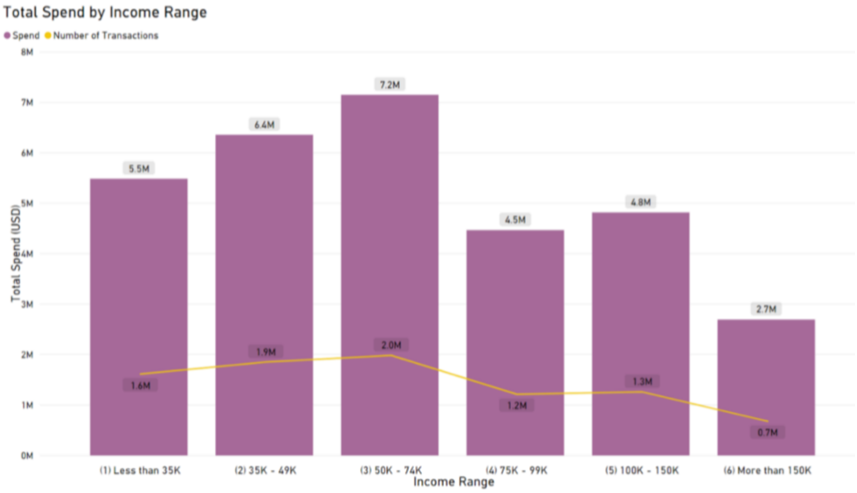


Figure 10

We noticed that the 50,000 to 74,000 dollars group spent the most on products, at 7.2 million dollars, as well as had the greatest number of transactions at 2.0 million. The next two categories were Less Than 35,000 and 35,000 to 49,000 dollars, for which there was a 5.5 million dollars and 6.4 million dollars respectively spent, and 1.6 million and 1.9 million transactions made, respectively.

We also analyzed this by week (*Figure 5*), and found that the 50,000 to 74,000 dollars group and the Less Than 35,000 and 35,000 to 49,000 dollars groups had the steadiest trends of slight increases, while the 75,000 to 99,000 dollars group had a larger increase of spending, over weeks, over the period of two years.

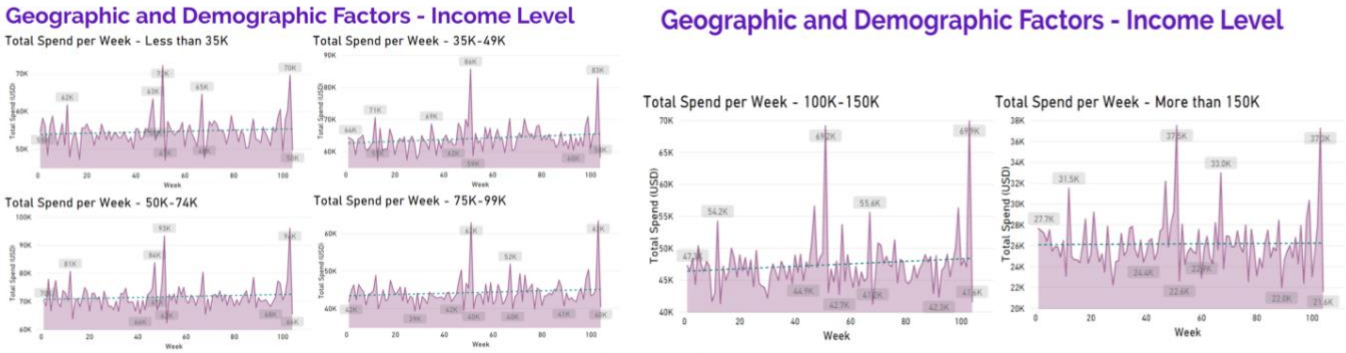
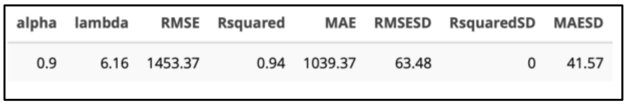


Figure 5

Predictive Modeling / forecasting. It was after extensive analysis that we accumulated all of this data and began to use Time Series Linear Models to predict potential patterns for future years and sales. We attempted several methods: a five-fold cross-validation, as well as linear regression, gradient boosted models, random forest, and elastic net models. We compared these root-mean-squared errors (RMSE) values, and found that the RMSE value of the elastic net model turned out to be the smallest. Therefore, we used this to pursue predictive modeling. The multiple R-squared of that model had a value of 0.94, which is rather strong. However, an RMSE of 1453.37 implies that the error for predicting the total spending for each household is approximately 1453.37, on average. In other words, on average, our prediction will likely be off target by about $1,453.37. It was the best possible outcome given our very narrow dataset. We additionally pursued elastic net regularization, and found that the RMSE was a difference between the values predicted by the model, and the values observed. Lastly, the multiple R-squared value was essentially the



alpha	lambda	RMSE	Rsquared	MAE	RMSESD	RsquaredSD	MAESD
0.9	6.16	1453.37	0.94	1039.37	63.48	0	41.57

percentage of variation in the response that was explained by the model. We break it down further, in the following description of our analysis.

Our first model was used to predict the total spending for each household. Before we fitted the model, we did some data cleaning. We created a new dataset with each household's information and its total spending in one year. We created a new column called "Region", for the households that made purchases only in one region, we put that region value in for the household, for the households that had purchases in multiple regions, we left those cells blank. We removed household number and all the "null" and "NA" values. We additionally removed some outliers based on Cook's distance. When fitting the model, we used 5-fold cross-validation, which randomly divided the observations into five parts. Each part was used as the test set alternately, and the remaining parts were used as a training set.

We used a linear model, a gradient boosted model, a random forest model, and an elastic net model to predict the total spending per year for a household. We found out that the elastic net model performed the best. Elastic net regularization is a penalized model which combines the Lasso and Ridge methods, it has a penalty for each variable added into the model.

We can see that the model fitted the model pretty well, the multiple R-squared is 0.94, which means that 94% of variation in total spending is explained by the model. However, when making predictions, the model did not perform so well, the root-mean-square error is 1453. In other words, on average, our prediction will be off for more than 14 hundred dollars. Some possible reasons are we only have 2 years of data for 5000 households, and households with a similar status may not have the same spending habit.

We calculated the mean weekly total spending and the standard deviation of weekly total spending, and we tried to see if there are any weeks that have total spending not within two standard deviations from the mean. As you can see from the graph, the two orange lines are two standard deviations from the mean. There are about 5 or 6 weeks that have total spending near or larger than the orange line above, which is how we found that these six weeks are Easter, Thanksgiving, and Christmas.

We chose to perform a linear time series model to forecast the total spending in the next 5 years because we did not know about the predictor variables in advance, so we forecasted sale based on calendar variables. The dark shaded region shows 80 percent prediction intervals and the light shaded region shows 95 percent prediction intervals. The value of prediction intervals is that they express the uncertainty in the forecasts. The trends showing in this plot is congruent with the regional plots that the west and east region have increasing trends while central and south remain constant. These are broken down into four graphs, for each region, South, West, Central, and East (*Figure 12*).

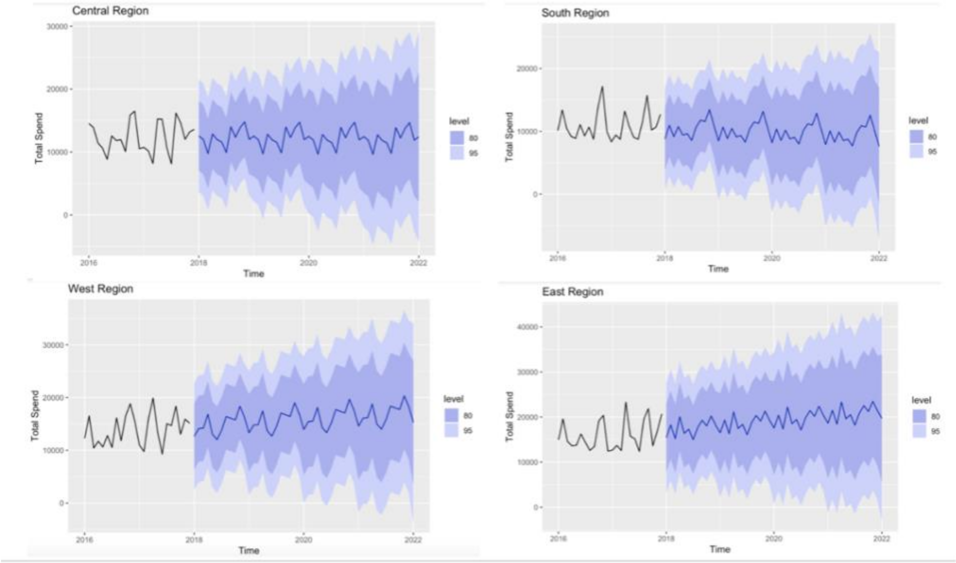


Figure 12

Anomalies and outliers. The time frame of our dataset made our group cautious of any of the conclusions we came to, in terms of suggestions for increasing future sales for Kroger; that said, we are confident of our decision to remain uncertain, as a project such as this needs multiple years, if not decades, worth of data. Knowing this, we decided to begin one final section, analyzing all possible anomalies and outliers that could have an effect on our suggestions, as well as on projections for future sales. We calculated the average total spending by weeks, as well as the standard deviation of the weekly total spending; using this, we attempted to discern if there were any weeks that consisted of the total spending not remaining within two standard deviations (SDs) from the average. Indeed, we found that the Easter, Thanksgiving, and Christmas weeks had total spending values that were near to or larger than two standard deviations from the average (*Figure 11*). Therefore, we dug into these six weeks' worth of data (three for each of the two years) and attempted to search for the commodities' sale that demonstrated a significant increase during these weeks. We found that during Easter, the Holiday commodity seemed to have a very significant increase in sales, as opposed to the rest of the six items shown in *Figure 11*. Furthermore, the increase was nearly three times greater than the second most purchased commodity. For the Thanksgiving week, our data showed that the Meat category was most purchased, particularly Turkey, by more than three times the second most purchased commodity, again. Finally, for the Christmas week, Holiday items once again were purchased the most, consistently, over the two-year period, and therefore, we suggest looking into increasing the stock of these products, during these three holidays.

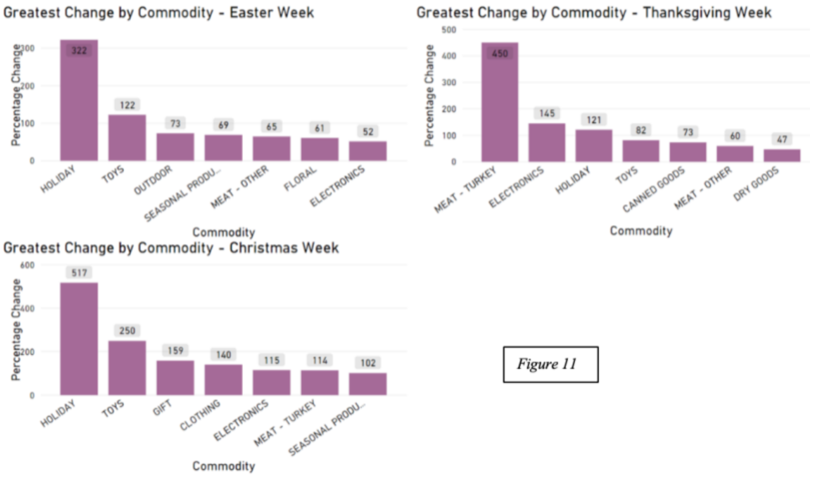


Figure 11

CONCLUSIONS

Customers to target. The following suggestions are with regards to our initial questions. To establish clarity, those questions were: “Based on the information from our data analyses, what are the current sales trends, and how can Kroger increase future sales? This included demographic and geographic factors, product performance, time series models, and looking at outliers and anomalies. 2) What are the limitations of the data we currently have available to us?”

In terms of the regions, we have noticed that the Central and the South regions needed to be targeted more. As for the income range, the customers earning less than $35,000 and the customers earning more than $150,000 are the two categories that need to be targeted far more than they currently are. Lastly, for the age groups, the group of 19 to 24 years old, and the group of 55 years old and above need to be targeted more. These are the customers that have decreased their expenditure, or have not increased their expenditures significantly between the years 2016 and 2017.

As for the groups and products that are doing well, the East region is doing very well, the age group of 45 to 54 is doing very well, and the age groups of 35-44 and 55-64 are doing well but also could be targeted a bit more. In terms of marital status, “Single” customers could be targeted more, and “Married” customers are doing well for Kroger. For Home Status, Kroger should target the Renters, and is doing well with the Home Owners. Finally, the $50,000 to $74,000 income range is most significantly helping sales with Kroger, and should continue to be targeted.

Limitations in dataset. In terms of the limitations to the dataset, for one, we are not sure when this customer data was collected. Additionally, there were only two years’ worth of data, making the fundamental base of this analysis a challenge, as we had to be very wary of the conclusions we were drawing, as well as of the analysis that we were understanding from the short time period. There was also very unclear data in terms of the Children being unclear, and there was missing information from this category that could not be differentiated. Lastly, we did not have the data available to us by Store, which would have dramatically increased our ability to dig deeper in this analysis. In order to explore further questions, we would need the above data, as well as potentially at least a decade's worth of updated data.

We want to thank Matt, Dave, and Professor Glosemeyer for the opportunity to work with 84.51 over the last couple months. We gained deep insight into the meaning of analyzing a dataset, working with a team, collaborating with a company, and producing and communicating data and results that has the potential to impact human lives. Appendices and Group Member Contributions are on the following pages.

***

APPENDICES

~~~
# Introduction


## Combine the data

```{r}
# Import data
households = read.csv("5000_households.csv")
products = read.csv("5000_products.csv")
transactions = read.csv("5000_transactions.csv")

# Creating quarters with respect to purchase dates
transactions$QUARTER = quarters(as.Date(transactions$PURCHASE_, format = "%d-%b-%y"))

# Merging datasets
data_merge1 = merge(transactions, products, by = "PRODUCT_NUM", all = FALSE)
data_merge2 = merge(transactions, households, by = "HSHD_NUM", all = FALSE)



```

```{r}
# Creating a new dataset 'spending'
spending = data.frame(
  "HSHD_NUM" = households$HSHD_NUM,
  "Spending_2016" = rep(0, 5000),
  "Spending_2017" = rep(0, 5000),
  "Change" = rep(0, 5000),
  "Trend" = rep(character(1), 5000)
  )

transactions_2016 = filter(data_merge2, YEAR == 2016)
transactions_2017 = filter(data_merge2, YEAR == 2017)

for (i in 1:5000) {
  spending$Spending_2016[i] = sum(transactions_2016$SPEND[transactions_2016$HSHD_NUM == spending$HSHD_NUM[i]])
  spending$Spending_2017[i] = sum(transactions_2017$SPEND[transactions_2017$HSHD_NUM == spending$HSHD_NUM[i]])
~~~

~~~~
  spending$Change[i] = spending$Spending_2017[i] - spending$Spending_2016[i]
  if (spending$Change[i] < 0) {
    spending$Trend = "decrease"
  } else {
    spending$Trend = "increase"
  }
}

```
~~~~

~~~~
```{r}
# Importing a dataset with all N/A values removed.
households_fixed = read.csv("5000_households_fixed.csv")

# Combining region with household information
households_region = summarise(group_by(transactions, HSHD_NUM), Uniqueness = isTRUE(length(unique(STORE_R)) == 1))
households_2 = merge(households_fixed, households_region, by = "HSHD_NUM", all = FALSE)
households_2$Region = rep(character(1), 5000)
Region = c("CENTRAL", "EAST", "SOUTH", "WEST")

for (i in 1:5000) {
  if (households_2$Uniqueness[i] == TRUE) {
    households_2$Region[i] = Region[unique(transactions$STORE_R[transactions$HSHD_NUM == households_2$HSHD_NUM[i]])]
  }
}

data_merge3 = merge(transactions, households_2, by = "HSHD_NUM", all = FALSE)

```
~~~~

~~~~
```{r warning = FALSE}
# Creating a dataset defined by basket number
Basket = summarise(group_by(data_merge3,BASKET_NUM,PURCHASE_,WEEK_NUM,YEAR,QUARTER,L,AGE_RANGE,MARITAL,INCOME_RANGE,HOMEOWNER,HSHD_COMPOSITION,HH_SIZE,CHILDREN,Uniqueness,Region),totalSPEND = sum(SPEND), totalUnit = sum(UNITS))

# Creating a dataset that aims to evaluate total spending of each household
~~~~

~~~
totalSpend = summarise(group_by(data_merge3,HSHD_NUM,L,AGE_RANGE,MARITAL,INCOME_RANGE,HOMEOWNER,HSHD_COMPOSITION,HH_SIZE,CHILDREN,Uniqueness,Region),totalSPEND = sum(SPEND), totalUnit = sum(UNITS))

# Removing HSHD_NUM and Uniqueness & Creating a Linear Model using totalSPEND as a function of all other variables
totalSpend$HSHD_NUM = NULL
totalSpend$Uniqueness = NULL
spend_mod = lm(totalSPEND ~., data = totalSpend)

```


```{r}
# Figuring and removing potential outliers
plot(spend_mod)
mod_cd = cooks.distance(spend_mod)
totalSpend_fix = totalSpend[mod_cd < 4 / length(mod_cd), ]
mod_fix = lm(totalSPEND ~ . , data = totalSpend_fix)
summary(mod_fix)


# Running stepwise selection to find out the major attribute to the dependent variable
step(mod_fix)

# Refitting the model
select_spend_mod = lm(
  formula = totalSPEND ~ AGE_RANGE + INCOME_RANGE + HOMEOWNER +
  HSHD_COMPOSITION + Region + totalUnit,
  data = totalSpend_fix
  )

# R2 for the original model vs. new model
data.frame(
  orig_r2 = summary(spend_mod)$adj.r.squared,
  model__r2 = summary(select_spend_mod)$adj.r.squared
  )

# Sorting the data
d2 = arrange(totalSpend_fix, totalSPEND)

# Performing Elastic Net
set.seed(8451)
mod_elastic = train(
  totalSPEND ~ . ,
  data = d2,
~~~

```
  trControl = trainControl(method = "cv", number = 5),
  method = "glmnet",
  tuneLength = 10
  )
```

```{r}
# Performing Random Forest
set.seed(8451)
spend_mod_rf = train(
  totalSPEND ~ .,
  data = d2,
  trControl = trainControl(method = "cv", number = 5),
  method = "rf",
  tuneGrid = expand.grid(mtry = seq(1, ncol(totalSpend_fix) - 1))
  )
```

```{r}
# Performing Gradient Boosting Machine
gbm_grid = expand.grid(interaction.depth = c(1, 2, 3),
                       n.trees = (1:30) * 100,
                       shrinkage = c(0.1, 0.3),
                       n.minobsinnode = c(10, 20))

set.seed(8451)
spend_mod_gbm = train(
  totalSPEND ~ .,
  data = d2,
  trControl = trainControl(method = "cv", number = 5),
  method = "gbm",
  tuneGrid = gbm_grid,
  verbose = FALSE
  )
```

```{r}

get_rmse = function(actual, predicted) {
  sqrt(mean((actual - predicted) ^ 2))
}
get_best_result = function(caret_fit) {
  best = which(rownames(caret_fit$results) == rownames(caret_fit$bestTune))
  best_result = caret_fit$results[best,]
````

~~~~
  rownames(best_result) = NULL
  best_result
}

(rmse_result = data.frame(
  Model = c("Elastic Net", "Gradient Boosting Machine", "Random Forest"),
  TestError = c(
  get_best_result(mod_elastic)$RMSE,
  get_best_result(spend_mod_gbm)$RMSE,
  get_best_result(spend_mod_rf)$RMSE
  )
  ))

# Check performance of the Elastic Net model
ts = sort(totalSpend_fix$totalSPEND)
par(mfrow = c(2,2))
plot(ts)
plot(fitted(mod_elastic),col = "orange")

```


```{r}
# Result table for the Elastic Net
kable(get_best_result(mod_elastic), "html", digits = 2) %>%
  kable_styling(bootstrap_options = "striped", full_width = FALSE)
```

```{r}
# Plots that shows the distribution of total spend of each household
plot(fitted(mod_fix), resid(mod_fix),col = 'dodgerblue',pch = 20,cex = 1.5, xlab = 'fitted', ylab = 'residuals')
abline(h = 0, lty = 2, col = 'darkorange', lwd = 2)
plot(ts)

```


```{r}
# Separating data by regions and calculating the weekly total spend of each region
week_spend = summarise(group_by(transactions,WEEK_NUM,STORE_R),totalSPEND = sum(SPEND), totalUnit = sum(UNITS))
total_week_spend = summarise(group_by(week_spend,WEEK_NUM), totalSPEND = sum(totalSPEND), totalUnit = sum(totalUnit))

week_central = week_spend[week_spend$STORE_R == "CENTRAL",]
week_east = week_spend[week_spend$STORE_R == "EAST   ",]
week_west = week_spend[week_spend$STORE_R == "WEST   ",]
~~~~

~~~
week_south = week_spend[week_spend$STORE_R == "SOUTH  ",]

# Calculating two standard deviations away from the mean of total spend for regions in the
U.S.
mean(week_central$totalSPEND) - 2 * sd(week_central$totalSPEND)
mean(week_central$totalSPEND) + 2 * sd(week_central$totalSPEND)
mean(week_east$totalSPEND) - 2 * sd(week_east$totalSPEND)
mean(week_east$totalSPEND) + 2 * sd(week_east$totalSPEND)
mean(week_west$totalSPEND) - 2 * sd(week_west$totalSPEND)
mean(week_west$totalSPEND) + 2 * sd(week_west$totalSPEND)
mean(week_south$totalSPEND) - 2 * sd(week_south$totalSPEND)
mean(week_south$totalSPEND) + 2 * sd(week_south$totalSPEND)

# Calculating two standard deviations away from the mean of total spend for U.S.level
mean(total_week_spend$totalSPEND) - 2 * sd(total_week_spend$totalSPEND)
mean(total_week_spend$totalSPEND) + 2 * sd(total_week_spend$totalSPEND)
```



```{r}
# Sorting data and grouping observations by regions & Calculating daily total spend
daily_central = transactions[transactions$STORE_R == "CENTRAL",]
d_central = summarise(group_by(daily_central,PURCHASE_,STORE_R),totalSPEND =
sum(SPEND))

daily_east = transactions[transactions$STORE_R == "EAST   ",]
d_east = summarise(group_by(daily_east,PURCHASE_,STORE_R),totalSPEND =
sum(SPEND))

daily_west = transactions[transactions$STORE_R == "WEST   ",]
d_west = summarise(group_by(daily_west,PURCHASE_,STORE_R),totalSPEND =
sum(SPEND))

daily_south = transactions[transactions$STORE_R == "SOUTH  ",]
d_south = summarise(group_by(daily_south,PURCHASE_,STORE_R),totalSPEND =
sum(SPEND))

d_central_sort = d_central[order(as.Date(d_central$PURCHASE_, format = "%d-%b-%y")),]
d_east_sort = d_east[order(as.Date(d_east$PURCHASE_, format = "%d-%b-%y")),]
d_west_sort = d_west[order(as.Date(d_west$PURCHASE_, format = "%d-%b-%y")),]
d_south_sort = d_south[order(as.Date(d_south$PURCHASE_, format = "%d-%b-%y")),]


```
~~~

~~~
```{r}
# Creating dataset for holiday seasons & checking percentage increase for each commodity
with respect to its mean
commodity_week = summarise(group_by(data_merge1,COMMODITY,WEEK_NUM),
totalspend = sum(SPEND))
pinduoduo = summarise(group_by(data_merge1,COMMODITY), totalsale = sum(SPEND))

week_12 = filter(commodity_week, WEEK_NUM == "12")
week_67 = filter(commodity_week, WEEK_NUM == "67")
week_easter = merge(week_12, week_67, by = "COMMODITY")
week_easter$avg_spend = (week_easter$totalspend.x + week_easter$totalspend.y) / 2
week_easter_final = merge(week_easter, pinduoduo, by = "COMMODITY")
week_easter_final$avg = week_easter_final$totalsale / 104
week_easter_final$pctdiff = (week_easter_final$avg_spend - week_easter_final$avg) /
week_easter_final$avg * 100

week_47 = filter(commodity_week, WEEK_NUM == "47")
week_99 = filter(commodity_week, WEEK_NUM == "99")
week_tg = merge(week_47, week_99, by = "COMMODITY")
week_tg$avg_spend = (week_tg$totalspend.x + week_tg$totalspend.y) / 2
week_tg_final = merge(week_tg, pinduoduo, by = "COMMODITY")
week_tg_final$avg = week_tg_final$totalsale / 104
week_tg_final$pctdiff = (week_tg_final$avg_spend - week_tg_final$avg) / week_tg_final$avg *
100

week_51 = filter(commodity_week, WEEK_NUM == "51")
week_103 = filter(commodity_week, WEEK_NUM == "103")
week_xmas = merge(week_51, week_103, by = "COMMODITY")
week_xmas$avg_spend = (week_xmas$totalspend.x + week_xmas$totalspend.y) / 2
week_xmas_final = merge(week_xmas, pinduoduo, by = "COMMODITY")
week_xmas_final$avg = week_xmas_final$totalsale / 104
week_xmas_final$pctdiff = (week_xmas_final$avg_spend - week_xmas_final$avg) /
week_xmas_final$avg * 100
```

```{r}
# Creating new datasets based on commondities
us_product = summarise(
  group_by(data_merge1, COMMODITY),
  totalSPEND = sum(SPEND),
  totalUnit = sum(UNITS)
  )
us_product = mutate(
  us_product,
  sale_percentage = totalSPEND / sum(totalSPEND) * 100,
  unit_percentage = totalUnit / sum(totalUnit) * 100
~~~

~~~
  )
west_product = summarise(
  group_by(data_merge1[data_merge1$STORE_R == "WEST   ", ], STORE_R,
COMMODITY),
  totalSPEND = sum(SPEND),
  totalUnit = sum(UNITS)
  )
west_product = mutate(
  west_product,
  sale_percentage = totalSPEND / sum(totalSPEND) * 100,
  unit_percentage = totalUnit / sum(totalUnit) * 100
  )
east_product = summarise(
  group_by(data_merge1[data_merge1$STORE_R == "EAST   ", ], STORE_R, COMMODITY),
  totalSPEND = sum(SPEND),
  totalUnit = sum(UNITS)
  )
east_product = mutate(
  east_product,
  sale_percentage = totalSPEND / sum(totalSPEND) * 100,
  unit_percentage = totalUnit / sum(totalUnit) * 100
  )
south_product = summarise(
  group_by(data_merge1[data_merge1$STORE_R == "SOUTH  ", ], STORE_R,
COMMODITY),
  totalSPEND = sum(SPEND),
  totalUnit = sum(UNITS)
  )
south_product = mutate(
  south_product,
  sale_percentage = totalSPEND / sum(totalSPEND) * 100,
  unit_percentage = totalUnit / sum(totalUnit) * 100
  )
central_product = summarise(
  group_by(data_merge1[data_merge1$STORE_R == "CENTRAL", ], STORE_R,
COMMODITY),
  totalSPEND = sum(SPEND),
  totalUnit = sum(UNITS)
  )
central_product = mutate(
  central_product,
  sale_percentage = totalSPEND / sum(totalSPEND) * 100,
  unit_percentage = totalUnit / sum(totalUnit) * 100
  )
```
~~~

~~~
```{r}
# Exporting datasets
write.csv(us_product, "us_product.csv")
write.csv(west_product, "west_product.csv")
write.csv(east_product, "east_product.csv")
write.csv(south_product, "south_product.csv")
write.csv(central_product, "central_product.csv")
```

```{r}
# Fitting a Linear Model with time series components
ts_spend = summarise(
  group_by(transactions, PURCHASE_),
  totalSPEND = sum(SPEND),
  totalUnit = sum(UNITS)
  )
ts_spend$PURCHASE_ = as.Date(ts_spend$PURCHASE_, format = '%d-%b-%y')
ts_spend = ts_spend[order(ts_spend$PURCHASE_),]
ts_spend$totalSPEND = as.numeric(ts_spend$totalSPEND)
str(ts_spend)
```

```{r}
# Test for Stationary
wt_fractdif = function(d, nwei,tau)
{
  wvec <- w0 <- 1
  if(is.null(tau))
  {
    for(k in 1:(nwei-1))
    {
      w1 = (-1)*w0*(d-k+1)/k
      wvec = c(wvec,w1)
      w0 = w1
    }
  }else
 {
    k = 1
    while(abs(w0) >= tau)
    {
      w1 = (-1)*w0*(d-k+1)/k
      wvec = c(wvec,w1)
      w0 = w1
      k = k+1
    }
~~~

~~~
    wvec = wvec[-length(wvec)]
 }
  return(wvec)
}

fracDiff = function(x, d, nwei, tau)
{
  weig = wt_fractdif(d = d, nwei = nwei, tau = tau)
  nwei = length(weig)
  nx = length(x)
  rst = sapply(nwei:nx,function(i){
    sum(weig*x[i:(i-nwei+1)])
  })
  return(rst)
}


fracD_c = fracDiff(ts_spend$totalSPEND, d = 0.5, tau = 0.001)
diff_c = diff(ts_spend)

trainDat <- ts_spend[1:floor(nrow(ts_spend)/3*2),]
testDat <- ts_spend[(floor(nrow(ts_spend)/3*2)+1):nrow(ts_spend),]

d_chosen <- 0
C_fracD <- fracDiff(trainDat$totalSPEND, d=d_chosen, tau=1e-4)
tseries::kpss.test(C_fracD, null="Trend")
tseries::adf.test(C_fracD)
tseries::pp.test(C_fracD)
stats::PP.test(C_fracD)
summary(urca::ur.ers(C_fracD, model = "trend"))
```

```{r}
ts_spend = mutate(ts_spend, MonthYear = paste(year(PURCHASE_), formatC(
  month(PURCHASE_), width = 2, flag = "0"
  )))

ts_monthly = aggregate(
  ts_spend$totalSPEND,
  by = list(ts_spend$MonthYear),
  FUN = function(x)
  mean(x, na.rm = T)
  )


myts = ts(
~~~

~~~~
ts_month$x,
frequency = 12,
start = c(2016, 01),
end = c(2017, 12)
)
plot(myts)

myds_monthly = decompose(myts)
plot(myds_monthly)

my_df_ts <- data.frame(totalspend = myts, as.numeric(time(myts)))
names(my_df_ts) = c("totalspend", "time")
mymodel = tslm(totalspend ~ season+ trend, my_df_ts)
my_fc = forecast(mymodel,h = 49,scientific = FALSE)


autoplot(my_fc, main = "US",ylab = "Total Spend")


```

```{r}
# Fitting a Linear Model with time series components (by week)
ts_spend = mutate(ts_spend, Week = paste(year(PURCHASE_),formatC(week(PURCHASE_),
width = 2, flag = "0")))

ts_week = aggregate(
  ts_spend$totalSPEND,
  by = list(ts_spend$Week),
  FUN = function(x)
  mean(x, na.rm = T)
  )

myts_week = ts(
  ts_week$x,
  frequency = 52,
  start = c(2016, 1),
  end = c(2017, 52)
  )

plot(myts_week)
myds_week = decompose(myts_week)
plot(myds_week)
my_df_ts_week = data.frame(totalspend = myts_week, as.numeric(time(myts_week)))
names(my_df_ts_week) = c("totalspend", "time")
mymodel_week = tslm(totalspend~season+trend,my_df_ts_week)
~~~~

~~~
my_fc_week = forecast(mymodel_week,h=220)
autoplot(my_fc_week,main = "US",ylab = "Total Spend")

```



```{r}
# Time series(central region_presentation)

d_central_sort_ts = d_central[order(as.Date(d_central$PURCHASE_, format = '%d-%b-%y')), ]
d_central_sort_ts$totalSPEND = as.numeric(d_central_sort_ts$totalSPEND)


central_ts = aggregate(
  d_central_sort_ts$totalSPEND,
  by = list(d_central_sort_ts$PURCHASE_),
  FUN = function(x)
  mean(x, na.rm = T)
  )

central_myts = ts(
  central_ts$x,
  frequency = 12,
  start = c(2016, 01),
  end = c(2017, 12)
  )

plot(central_myts)
myds_monthly_central = decompose(central_myts)
plot(myds_monthly_central)


my_df_ts_central = data.frame(totalspend = central_myts, as.numeric(time(central_myts)))
names(my_df_ts_central) = c("totalspend", "time")
mymodel_central = tslm(totalspend ~ season+ trend, my_df_ts_central)
my_fc_central = forecast(mymodel_central,h = 49)
autoplot(my_fc_central, main = "Central Region",ylab = "Total Spend")

```

```{r}
# Updated version: forecasting monthly total spend in central region
d_central_a = d_central_sort
d_central_a$PURCHASE_ = as.Date(d_central$PURCHASE_,format = '%d-%b-%y')
d_central_a = d_central_a[order(d_central_a$PURCHASE_),]
d_central_a$totalSPEND = as.numeric(d_central_a$totalSPEND)
~~~

~~~
str(d_central_a)

d_central_a = mutate(d_central_a, YearMonth = paste(year(PURCHASE_), formatC(
  month(PURCHASE_), width = 2, flag = "0"
  )))

central_tsa = aggregate(
  d_central_a$totalSPEND,
  by = list(d_central_a$YearMonth),
  FUN = function(x)
  mean(x, na.rm = T)
  )



central_mytsa = ts(
  central_tsa$x,
  frequency = 12,
  start = c(2016, 01, 03),
  end = c(2017, 12, 31)
  )
plot(central_mytsa)
myds_monthly_centrala = decompose(central_mytsa)
plot(myds_monthly_centrala)



my_df_ts_centrala = data.frame(totalspend = central_mytsa, as.numeric(time(central_mytsa)))
names(my_df_ts_centrala) = c("totalspend", "time")
mymodel_centrala = tslm(totalspend ~ season+ trend, my_df_ts_centrala)
my_fc_centrala = forecast(mymodel_centrala,h = 49)
autoplot(my_fc_centrala, main = "Central Region",ylab = "Total Spend")
```

```{r}
# Time series(west region_presentation)

d_west_sort_ts = d_west[order(as.Date(d_west$PURCHASE_,format = '%d-%b-%y')),]
d_west_sort_ts$totalSPEND = as.numeric(d_west_sort_ts$totalSPEND)

west_ts_monthly = aggregate(
  d_west_sort_ts$totalSPEND,
  by = list(d_west_sort_ts$PURCHASE_),
  FUN = function(x)
  mean(x, na.rm = T)
  )
~~~

~~~
west_myts = ts(
  west_ts_monthly$x,
  frequency = 12,
  start = c(2016, 01),
  end = c(2017, 12)
  )
plot(west_myts)

myds_monthly_west = decompose(west_myts)
plot(myds_monthly_west)


my_df_ts_west = data.frame(totalspend = west_myts, as.numeric(time(west_myts)))
names(my_df_ts_west) = c("totalspend", "time")
mymodel_west = tslm(totalspend ~ season+ trend, my_df_ts_west)
my_fc_west = forecast(mymodel_west,h = 49)
autoplot(my_fc_west, main = "West Region",ylab = "Total Spend")
```
```{r}
# Updated version: forecasting monthly total spend in west region
d_west_a = d_west_sort
d_west_a$PURCHASE_ = as.Date(d_west$PURCHASE_,format = '%d-%b-%y')
d_west_a = d_west_a[order(d_west_a$PURCHASE_),]
d_west_a$totalSPEND = as.numeric(d_west_a$totalSPEND)
str(d_west_a)

d_west_a = mutate(d_west_a, YearMonth =
paste(year(PURCHASE_),formatC(month(PURCHASE_), width = 2, flag = "0")))

west_tsa_monthly = aggregate(
  d_west_a$totalSPEND,
  by = list(d_west_a$YearMonth),
  FUN = function(x)
  mean(x, na.rm = T)
  )

west_mytsa = ts(
  west_tsa_monthly$x,
  frequency = 12,
  start = c(2016, 01),
  end = c(2017, 12)
  )
plot(west_mytsa)

mydsa_monthly_west = decompose(west_myts)
plot(mydsa_monthly_west)
~~~

~~~
my_df_tsa_west = data.frame(totalspend = west_mytsa, as.numeric(time(west_mytsa)))
names(my_df_tsa_west) = c("totalspend", "time")
mymodela_west = tslm(totalspend ~ season+ trend, my_df_tsa_west)
my_fca_west = forecast(mymodela_west,h = 49)
autoplot(my_fca_west, main = "West Region",ylab = "Total Spend")
```

```{r}
# Time series(east region)
d_east_sort_ts = d_east[order(as.Date(d_east$PURCHASE_, format = "%d-%b-%y")),]
d_east_sort_ts$totalSPEND = as.numeric(d_east_sort_ts$totalSPEND)

east_ts_monthly = aggregate(
  d_east_sort_ts$totalSPEND,
  by = list(d_east_sort_ts$PURCHASE_),
  FUN = function(x)
  mean(x, na.rm = T)
  )

east_myts = ts(
  east_ts_monthly$x,
  frequency = 12,
  start = c(2016, 01),
  end = c(2017, 12)
  )
plot(east_myts)

myds_monthly_east = decompose(east_myts)
plot(myds_monthly_east)


my_df_ts_east = data.frame(totalspend = east_myts, as.numeric(time(east_myts)))
names(my_df_ts_east) = c("totalspend", "time")
mymodel_east = tslm(totalspend ~ season+ trend, my_df_ts_east)
my_fc_east = forecast(mymodel_east,h = 49)
autoplot(my_fc_east, main = "East Region",ylab = "Total Spend")
```
```{r}
# Updated version for east region
d_east_a = d_east_sort
d_east_a$PURCHASE_ = as.Date(d_east$PURCHASE_,format = '%d-%b-%y')
d_east_a = d_east_a[order(d_east_a$PURCHASE_),]
d_east_a$totalSPEND = as.numeric(d_east_a$totalSPEND)
str(d_east_a)
~~~

~~~
d_east_a = mutate(d_east_a, YearMonth =
paste(year(PURCHASE_),formatC(month(PURCHASE_), width = 2, flag = "0")))

east_tsa_monthly = aggregate(
  d_east_a$totalSPEND,
  by = list(d_east_a$YearMonth),
  FUN = function(x)
  mean(x, na.rm = T)
  )

east_mytsta = ts(
  east_tsa_monthly$x,
  frequency = 12,
  start = c(2016, 01),
  end = c(2017, 12)
  )
  plot(east_mytsta)

mydsa_monthly_east = decompose(east_mytsta)
plot(mydsa_monthly_east)


my_df_tsa_east = data.frame(totalspend = east_mytsta, as.numeric(time(east_mytsta)))
names(my_df_tsa_east) = c("totalspend", "time")
mymodela_east = tslm(totalspend ~ season+ trend, my_df_tsa_west)
my_fca_east = forecast(mymodela_east,h = 49)
autoplot(my_fca_east, main = "East Region",ylab = "Total Spend")
```

```{r}
# Time series(south region)
d_south_sort_ts = d_south[order(as.Date(d_south$PURCHASE_, format = "%d-%b-%y")),]
d_south_sort_ts$totalSPEND = as.numeric(d_south_sort_ts$totalSPEND)

south_ts_monthly = aggregate(
  d_south_sort_ts$totalSPEND,
  by = list(d_south_sort_ts$PURCHASE_),
  FUN = function(x)
  mean(x, na.rm = T)
  )

south_myts = ts(
  south_ts_monthly$x,
  frequency = 12,
  start = c(2016, 01),
~~~

~~~
  end = c(2017, 12)
  )
  plot(south_myts)

myds_monthly_south = decompose(south_myts)
plot(myds_monthly_south)


my_df_ts_south = data.frame(totalspend = south_myts, as.numeric(time(south_myts)))
names(my_df_ts_south) = c("totalspend", "time")
mymodel_south = tslm(totalspend ~ season+ trend, my_df_ts_south)
my_fc_south = forecast(mymodel_south,h = 49)
autoplot(my_fc_south, main = "South Region", ylab = "Total Spend")
```

```{r}
# Updated time series(south region)
d_south_a = d_south_sort
d_south_a$PURCHASE_ = as.Date(d_east$PURCHASE_,format = '%d-%b-%y')
d_south_a = d_south_a[order(d_south_a$PURCHASE_),]
d_south_a$totalSPEND = as.numeric(d_south_a$totalSPEND)
str(d_south_a)

d_south_a = mutate(d_south_a, YearMonth =
paste(year(PURCHASE_),formatC(month(PURCHASE_), width = 2, flag = "0")))

south_tsa_monthly = aggregate(
  d_south_a$totalSPEND,
  by = list(d_south_a$YearMonth),
  FUN = function(x)
  mean(x, na.rm = T)
  )

south_mytsta = ts(
  south_tsa_monthly$x,
  frequency = 12,
  start = c(2016, 01),
  end = c(2017, 12)
  )

plot(south_mytsta)

mydsa_monthly_south = decompose(south_mytsta)
plot(mydsa_monthly_south)
~~~

~~~
my_df_tsa_south <- data.frame(totalspend = south_mytsta, as.numeric(time(south_mytsta)))
names(my_df_tsa_south) <- c("totalspend", "time")
mymodela_south = tslm(totalspend ~ season+ trend, my_df_tsa_south)
my_fca_south = forecast(mymodela_south,h = 49)
autoplot(my_fca_south, main = "South Region",ylab = "Total Spend")
```



```{r}
# Information related to week 12
wk_12 = summarise(group_by(data_merge1,COMMODITY,WEEK_NUM), totalspend =
sum(SPEND))
week_12 = filter(wk_12, WEEK_NUM == '12')
pinduoduo = summarise(group_by(data_merge1,COMMODITY), totalsale = sum(SPEND))
week_12_final = merge(week_12, pinduoduo, by = "COMMODITY")
week_12_final$avg = week_12_final$totalsale / 104
week_12_final$diff = (week_12_final$totalspend - week_12_final$avg) / week_12_final$avg
```

```{r}
# Evaluating relationship between income level and comodities
demo_data = merge(data_merge2,products, by = "PRODUCT_NUM")
demo_product_income =
summarise(group_by(demo_data,COMMODITY,INCOME_RANGE),totalSPEND =
sum(SPEND))
pp1 = summarise(group_by(demo_data,INCOME_RANGE), totalsale = sum(SPEND))
demo_product_income = merge(demo_product_income, pp1, by = "INCOME_RANGE")
demo_product_income = mutate(demo_product_income, percent = totalSPEND/totalsale*100)
write.csv(demo_product_income,"demo_product_income")
```



```{r}
# Evaluating relationship between marital status and commondities
demo_product_marital = summarise(group_by(demo_data, COMMODITY, MARITAL),
totalSPEND = sum(SPEND))
pp2 = summarise(group_by(demo_data,MARITAL), totalsale = sum(SPEND))
demo_product_marital = merge(demo_product_marital, pp2, by = "MARITAL")
demo_product_marital = mutate(demo_product_marital, percent = totalSPEND/totalsale*100)
write.csv(demo_product_marital,"demo_product_marital")
```

```{r}
# Evaluating relationship between commodities and age
demo_product_age =
summarise(group_by(demo_data,COMMODITY,AGE_RANGE),totalSPEND = sum(SPEND))
~~~

~~~
pp3 = summarise(group_by(demo_data,AGE_RANGE), totalsale = sum(SPEND))
demo_product_age = merge(demo_product_age, pp3, by = "AGE_RANGE")
demo_product_age = mutate(demo_product_age, percent = totalSPEND/totalsale*100)
write.csv(demo_product_age,"demo_product_age")
```

```{r}
# Evaluating relationship between home status and amount spent on each commodities
demo_product_home =
summarise(group_by(demo_data,COMMODITY,HOMEOWNER),totalSPEND =
sum(SPEND))
pp4 = summarise(group_by(demo_data,HOMEOWNER), totalsale = sum(SPEND))
demo_product_home = merge(demo_product_home, pp4, by = "HOMEOWNER")
demo_product_home = mutate(demo_product_home, percent = totalSPEND/totalsale*100)
write.csv(demo_product_home,"demo_product_home")
```

```{r}
# 1500 cutoff, spliting the data into two: total spend < 3500 and total spend >= 3500
totalSpend_low = totalSpend[totalSpend$totalSPEND < 3500, ]
totalSpend_low$HSHD_NUM=NULL


spend_low = lm(totalSPEND~., data = totalSpend_low)
mod_cd_low = cooks.distance(spend_low)
totalSpend_low = totalSpend_low[mod_cd_low < 4 / length(mod_cd_low),]

set.seed(8451)

totalSpend_low_rf = train(
  totalSPEND ~ .,
  data = totalSpend_low,
  trControl = trainControl(method = "cv", number = 5),
  method = "rf",
  tuneGrid = expand.grid(mtry = seq(1, ncol(totalSpend_low) - 1)))


gbm_grid = expand.grid(interaction.depth = c(1, 2, 3),
                       n.trees = (1:30) * 100,
                       shrinkage = c(0.1, 0.3),
                       n.minobsinnode = c(10, 20))

set.seed(8451)
totalSpend_low_gbm = train(
  totalSPEND ~ .,
  data = totalSpend_low,
~~~

~~~
  trControl = trainControl(method = "cv", number = 5),
  method = "gbm",
  tuneGrid = gbm_grid,
  verbose = FALSE
)

set.seed(8451)
mod_elastic_low = train(
  totalSPEND ~ .,
  data = totalSpend_low,
  trControl = trainControl(method = "cv", number = 5),
  method = "glmnet",
  tuneLength = 10
)

get_best_result(mod_elastic_low)$RMSE
get_best_result(totalSpend_low_gbm)$RMSE
get_best_result(totalSpend_low_rf)$RMSE
```

```{r}
totalSpend_1500 = totalSpend[totalSpend$totalSPEND >= 3500, ]
totalSpend_1500$HSHD_NUM = NULL

spend_1500 = lm(totalSPEND ~ ., data = totalSpend_1500)
mod_cd_1500 = cooks.distance(spend_1500)
totalSpend_1500_fix = totalSpend_1500[mod_cd_1500 < 4 / length(mod_cd_1500), ]

set.seed(8451)
totalSpend_1500_rf = train(
  totalSPEND ~ .,
  data = totalSpend_1500_fix,
  trControl = trainControl(method = "cv", number = 5),
  method = "rf",
  tuneGrid = expand.grid(mtry = seq(1, 2*ncol(totalSpend_1500_fix) - 1)))

set.seed(8451)
totalSpend_1500_gbm = train(
  totalSPEND ~ .,
  data = totalSpend_1500_fix,
  trControl = trainControl(method = "cv", number = 5),
  method = "gbm",
  tuneGrid = gbm_grid,
  verbose = FALSE
)
~~~

~~~
set.seed(8451)
mod_elastic_1500 = train(
  totalSPEND ~ .,
  data = totalSpend_1500,
  trControl = trainControl(method = "cv", number = 5),
  method = "glmnet",
  tuneLength = 10
)

get_best_result(mod_elastic_1500)$RMSE
get_best_result(totalSpend_1500_gbm)$RMSE
get_best_result(totalSpend_1500_rf)$RMSE
totalSpend_1500_rf$results

```
~~~

GROUP MEMBER CONTRIBUTIONS

Mid-Project Check-In

Power BI visualizations: Siddharth Ahuja

R-code: Zhe Huang, Jiewen Wu

Mid-Project Slides: Siddharth Ahuja, Zhe Huang, Krti Tallam, Jiewen Wu

Analysis: Siddharth Ahuja, Krti Tallam, Zhe Huang, Jiewen Wu

Mid-Project Report: Krti Tallam

Final Project

Power BI visualizations: Siddharth Ahuja

R-code: Zhe Huang, Jiewen Wu, Krti Tallam

Final Slides: Siddharth Ahuja with assistance from Zhe Huang, Jiewen Wu, Krti Tallam

Analysis: Jiewen Wu, Krti Tallam, Siddharth Ahuja, Zhe Huang

Final Report: Krti Tallam, with peer-review from Jiewen Wu, Zhe Huang, Siddharth Ahuja