Black Friday

15/01/2019

**Exploratory Data Analysis (EDA)**

To begin, lets load the dataset that we wil be using for this Exploratory Data Analysis (EDA).

dataset = read.csv("BlackFriday.csv")

Now, lets import the libraries we will be utilizing in this kernel.

library(tidyverse)

## ── Attaching packages ────────────────────────────────────────────────────────────────────── tidyverse 1.2.1 ──

## ✔ ggplot2 3.1.0 ✔ purrr 0.2.5  
## ✔ tibble 2.0.1 ✔ dplyr 0.7.8  
## ✔ tidyr 0.8.2 ✔ stringr 1.3.1  
## ✔ readr 1.3.1 ✔ forcats 0.3.0

## Warning: package 'tibble' was built under R version 3.5.2

## ── Conflicts ───────────────────────────────────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()

library(scales)

##   
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':  
##   
## discard

## The following object is masked from 'package:readr':  
##   
## col\_factor

library(arules)

## Loading required package: Matrix

##   
## Attaching package: 'Matrix'

## The following object is masked from 'package:tidyr':  
##   
## expand

##   
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

library(gridExtra)

##   
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':  
##   
## combine

The tidyverse package is what we will use for visualizing and exploring our dataset. It is knows for easy-to-read syntax and massive amounts of useful functions. The scales package will be used mainly to customize plot axis. Lastly, the arules package will be utilized in the final part of this kernel, Association Rule Learning and Apriori. Info regarding all packages used during this EDA is provided in the Works Cited section of this kernel.

Lets start with a quick overview of the entire dataset.

summary(dataset)

## User\_ID Product\_ID Gender Age   
## Min. :1000001 P00265242: 1858 F:132197 0-17 : 14707   
## 1st Qu.:1001495 P00110742: 1591 M:405380 18-25: 97634   
## Median :1003031 P00025442: 1586 26-35:214690   
## Mean :1002992 P00112142: 1539 36-45:107499   
## 3rd Qu.:1004417 P00057642: 1430 46-50: 44526   
## Max. :1006040 P00184942: 1424 51-55: 37618   
## (Other) :528149 55+ : 20903   
## Occupation City\_Category Stay\_In\_Current\_City\_Years  
## Min. : 0.000 A:144638 0 : 72725   
## 1st Qu.: 2.000 B:226493 1 :189192   
## Median : 7.000 C:166446 2 : 99459   
## Mean : 8.083 3 : 93312   
## 3rd Qu.:14.000 4+: 82889   
## Max. :20.000   
##   
## Marital\_Status Product\_Category\_1 Product\_Category\_2 Product\_Category\_3  
## Min. :0.0000 Min. : 1.000 Min. : 2.00 Min. : 3.0   
## 1st Qu.:0.0000 1st Qu.: 1.000 1st Qu.: 5.00 1st Qu.: 9.0   
## Median :0.0000 Median : 5.000 Median : 9.00 Median :14.0   
## Mean :0.4088 Mean : 5.296 Mean : 9.84 Mean :12.7   
## 3rd Qu.:1.0000 3rd Qu.: 8.000 3rd Qu.:15.00 3rd Qu.:16.0   
## Max. :1.0000 Max. :18.000 Max. :18.00 Max. :18.0   
## NA's :166986 NA's :373299   
## Purchase   
## Min. : 185   
## 1st Qu.: 5866   
## Median : 8062   
## Mean : 9334   
## 3rd Qu.:12073   
## Max. :23961   
##

head(dataset)

## User\_ID Product\_ID Gender Age Occupation City\_Category  
## 1 1000001 P00069042 F 0-17 10 A  
## 2 1000001 P00248942 F 0-17 10 A  
## 3 1000001 P00087842 F 0-17 10 A  
## 4 1000001 P00085442 F 0-17 10 A  
## 5 1000002 P00285442 M 55+ 16 C  
## 6 1000003 P00193542 M 26-35 15 A  
## Stay\_In\_Current\_City\_Years Marital\_Status Product\_Category\_1  
## 1 2 0 3  
## 2 2 0 1  
## 3 2 0 12  
## 4 2 0 12  
## 5 4+ 0 8  
## 6 3 0 1  
## Product\_Category\_2 Product\_Category\_3 Purchase  
## 1 NA NA 8370  
## 2 6 14 15200  
## 3 NA NA 1422  
## 4 14 NA 1057  
## 5 NA NA 7969  
## 6 2 NA 15227

It looks like we have 12 different columns, each representing a corresponding variable below.

* User\_ID: Unique identifier of shopper.
* Product\_ID: Unique identifier of product. (No key given)
* Gender: Sex of shopper.
* Age: Age of shopper split into bins.
* Occupation: Occupation of shopper. (No key given)
* City\_Category: Residence location of shopper. (No key given)
* Stay\_In\_Current\_City\_Years: Number of years stay in current city.
* Marital\_Status: Marital status of shopper.
* Product\_Category\_1: Product category of purchase.
* Product\_Category\_2: Product may belong to other category.
* Product\_Category\_3: Product may belong to other category.
* Purchase: Purchase amount in dollars.

If we look at the first few rows of our dataset, we can see that each row represents a different transaction, or item purchased by a specific customer. This will come into play later on when we group all transactions by a specific User\_ID to get a sum of all purchases made by a single customer.

One critique we can make regarding this dataset is that there isn’t a key given regarding the different Product\_IDs and the item they represent. (Ie. We can’t attribute P00265242 to an item easily recognizable) In reality, we would want to have another dataset which provides the name of an Item and its Product\_ID and then join it to our existing dataset. This won’t necessarily affect our EDA, but would be more useful during our implementation of the Apriori algorithm and could make some parts of the EDA clearer to interpret.

**Gender**

To begin our exploration, lets examine the gender of shoppers at this store.

Since each row represents an individual transaction, we must first group the data by User\_ID to remove duplicates.

dataset\_gender = dataset %>%  
 select(User\_ID, Gender) %>%  
 group\_by(User\_ID) %>%  
 distinct()  
head(dataset\_gender)

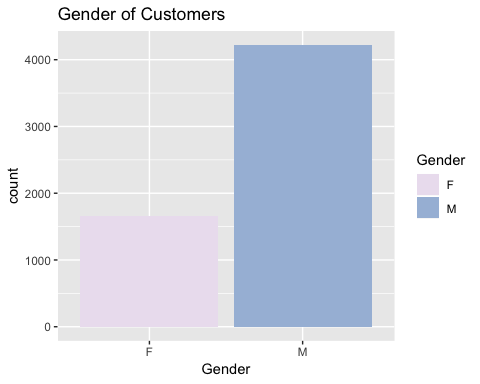
## # A tibble: 6 x 2  
## # Groups: User\_ID [6]  
## User\_ID Gender  
## <int> <fct>   
## 1 1000001 F   
## 2 1000002 M   
## 3 1000003 M   
## 4 1000004 M   
## 5 1000005 M   
## 6 1000006 F

summary(dataset\_gender$Gender)

## F M   
## 1666 4225

Now that we have the dataframe necessary to see each User\_IDs corresponding gender and their total counts for reference, lets plot the distribution of gender across our dataset.

options(scipen=10000) # To remove scientific numbering  
genderDist = ggplot(data = dataset\_gender) +  
 geom\_bar(mapping = aes(x = Gender, y = ..count.., fill = Gender)) +  
 labs(title = 'Gender of Customers') +   
 scale\_fill\_brewer(palette = 'PuBuGn')  
print(genderDist)



As we can see, there are quite a few more males than females shopping at our store on Black Friday. This gender split metric is helpful to retailers because some might want to modify their store layout, product selection, and other variables differently depending on the gender proportion of their shoppers.

A study published in the Clothing and Textiles Research Journal writes,

“Involvement, variety seeking, and physical environment of stores were selected as antecedents of shopping experience satisfaction….The structural model for female subjects confirmed the existence of the mediating role of hedonic shopping value in shopping satisfaction, whereas the model for male respondents did not.” Chang, E., Burns, L. D., & Francis, S. K. (2004) (Abstract) Although this does not give direct insight into recommended actions for retail stores, it does display a difference in the value derived from shopping and its relationship to gender, which should be taken into account by retailers.

To investigate further, lets compute the average spending amount as it relates to Gender. For easy interpretation and traceback we will create separate tables and then join them together.

total\_purchase\_user = dataset %>%  
 select(User\_ID, Gender, Purchase) %>%  
 group\_by(User\_ID) %>%  
 arrange(User\_ID) %>%  
 summarise(Total\_Purchase = sum(Purchase))  
user\_gender = dataset %>%  
 select(User\_ID, Gender) %>%  
 group\_by(User\_ID) %>%  
 arrange(User\_ID) %>%  
 distinct()  
head(user\_gender)

## # A tibble: 6 x 2  
## # Groups: User\_ID [6]  
## User\_ID Gender  
## <int> <fct>   
## 1 1000001 F   
## 2 1000002 M   
## 3 1000003 M   
## 4 1000004 M   
## 5 1000005 M   
## 6 1000006 F

head(total\_purchase\_user)

## # A tibble: 6 x 2  
## User\_ID Total\_Purchase  
## <int> <int>  
## 1 1000001 333481  
## 2 1000002 810353  
## 3 1000003 341635  
## 4 1000004 205987  
## 5 1000005 821001  
## 6 1000006 379450

user\_purchase\_gender = full\_join(total\_purchase\_user, user\_gender, by = "User\_ID")  
head(user\_purchase\_gender)

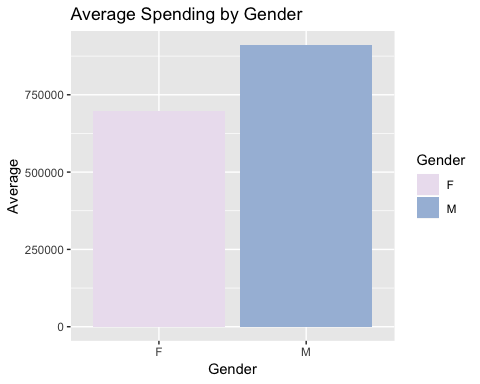
## # A tibble: 6 x 3  
## User\_ID Total\_Purchase Gender  
## <int> <int> <fct>   
## 1 1000001 333481 F   
## 2 1000002 810353 M   
## 3 1000003 341635 M   
## 4 1000004 205987 M   
## 5 1000005 821001 M   
## 6 1000006 379450 F

average\_spending\_gender = user\_purchase\_gender %>%  
 group\_by(Gender) %>%  
 summarize(Purchase = sum(as.numeric(Total\_Purchase)),   
 Count = n(),   
 Average = Purchase/Count)  
head(average\_spending\_gender)

## # A tibble: 2 x 4  
## Gender Purchase Count Average  
## <fct> <dbl> <int> <dbl>  
## 1 F 1164624021 1666 699054.  
## 2 M 3853044357 4225 911963.

We can see that that the average transaction for Females was 699054.00 and the average transaction for Males was 911963.20. Let visualize our results.

genderAverage = ggplot(data = average\_spending\_gender) +  
 geom\_bar(mapping = aes(x = Gender, y = Average, fill = Gender), stat = 'identity') +  
 labs(title = 'Average Spending by Gender') +  
 scale\_fill\_brewer(palette = 'PuBuGn')  
print(genderAverage)



Here we see an interesting observation. Even though female shoppers make less purchases than males at this specific store, they seem to be purchasing almost as much on average as the male shoppers. This being said, scale needs to be taken into account because females on average are still spending about 250,000 less than males.

\*\* Top Sellers \*\*

Now lets switch gears and examine our top selling products. In this situation, we won’t group by product ID since we want to see duplicates, just in case people are buying 2 or more quantities of the same product.

top\_sellers = dataset %>%  
 count(Product\_ID, sort = TRUE)  
top\_5 = head(top\_sellers, 5)  
top\_5

## # A tibble: 5 x 2  
## Product\_ID n  
## <fct> <int>  
## 1 P00265242 1858  
## 2 P00110742 1591  
## 3 P00025442 1586  
## 4 P00112142 1539  
## 5 P00057642 1430

Looks like our top 5 best sellers are (by product ID)

* P00265242 = 1858
* P00110742 = 1591
* P00025442 = 1586
* P00112142 = 1539
* P00057642 = 1430

Now that we have Identified our top 5 best selling products, lets examine the best selling product, P00265242.

best\_seller = dataset[dataset$Product\_ID == 'P00265242', ]  
head(best\_seller)

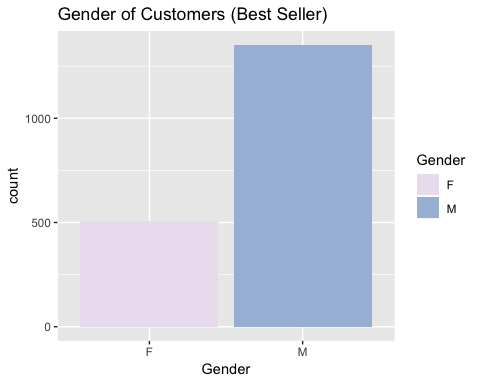
## User\_ID Product\_ID Gender Age Occupation City\_Category  
## 400 1000066 P00265242 M 26-35 18 C  
## 1192 1000196 P00265242 F 36-45 9 C  
## 1373 1000222 P00265242 M 26-35 1 A  
## 1846 1000301 P00265242 M 18-25 4 B  
## 2210 1000345 P00265242 M 26-35 12 A  
## 2405 1000383 P00265242 F 26-35 7 A  
## Stay\_In\_Current\_City\_Years Marital\_Status Product\_Category\_1  
## 400 2 0 5  
## 1192 4+ 0 5  
## 1373 1 0 5  
## 1846 4+ 0 5  
## 2210 2 1 5  
## 2405 4+ 1 5  
## Product\_Category\_2 Product\_Category\_3 Purchase  
## 400 8 NA 8652  
## 1192 8 NA 8767  
## 1373 8 NA 6944  
## 1846 8 NA 8628  
## 2210 8 NA 8593  
## 2405 8 NA 6998

We can see that this product fits into Product\_Category\_1 = 5 and Product\_Category\_2 = 8. As mentioned in the introduction, it would be useful to have a key to reference the item name in order to determine what it is.

Another interesting finding is that even though people are purchasing the same product, they are paying different prices. This could be due to various Black Friday promotions, discounts, or coupon codes. Otherwise, investigation would need to be done regarding the reason for different purchase prices of the same product between customers.

Lets continue to analyze our best seller to see if any relationship to Gender exits.

genderDist\_bs = ggplot(data = best\_seller) +  
 geom\_bar(mapping = aes(x = Gender, y = ..count.., fill = Gender)) +  
 labs(title = 'Gender of Customers (Best Seller)') +  
 scale\_fill\_brewer(palette = 'PuBuGn')  
print(genderDist\_bs)



We see a similar distribution between genders to our overall dataset gender split - lets confirm.

genderDist\_bs\_prop = ggplot(data = best\_seller) +   
 geom\_bar(fill = 'lightblue', mapping = aes(x = Gender, y = ..prop.., group = 1, fill = Gender)) +  
 labs(title = 'Gender of Customers (Best Seller - Proportion)') +  
 theme(plot.title = element\_text(size=9.5))  
genderDist\_prop = ggplot(data = dataset\_gender) +   
 geom\_bar(fill = "lightblue4", mapping = aes(x = Gender, y = ..prop.., group = 1)) +  
 labs(title = 'Gender of Customers (Total Proportion)') +  
 theme(plot.title = element\_text(size=9.5))   
grid.arrange(genderDist\_prop, genderDist\_bs\_prop, ncol=2)



We can see that between the overall observation set, both purchasers of the best seller and purchasers of all products are roughly ~25% female and ~75% male. A slight difference does exist but it seems like we can generally conclude that our best seller does not cater to a specific gender.

Now, let’s move on and examine the Age variable.

\*\* Age \*\*

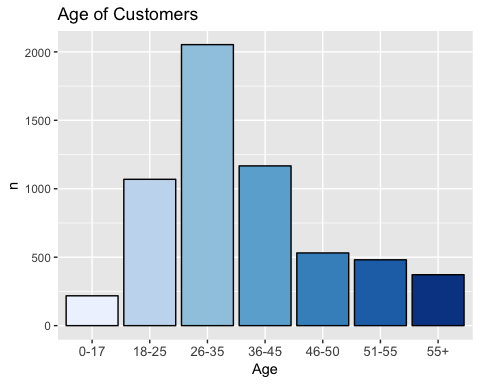
Lets begin examining Age by creating a table of each individual age group and their respective counts.

customers\_age = dataset %>%  
 select(User\_ID, Age) %>%  
 distinct() %>%  
 count(Age)  
customers\_age

## # A tibble: 7 x 2  
## Age n  
## <fct> <int>  
## 1 0-17 218  
## 2 18-25 1069  
## 3 26-35 2053  
## 4 36-45 1167  
## 5 46-50 531  
## 6 51-55 481  
## 7 55+ 372

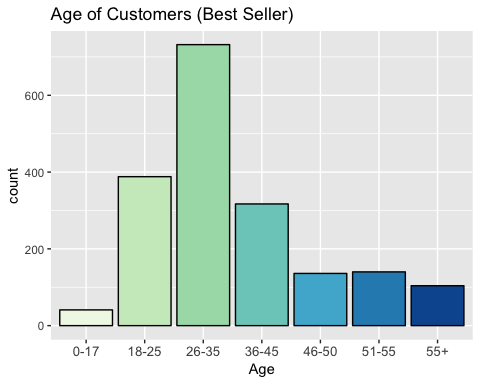
Here, we can see a dataset that shows the count of each Age category of customers at our store. Lets visualize this table.

customers\_age\_vis = ggplot(data = customers\_age) +   
 geom\_bar(color = 'black', stat = 'identity', mapping = aes(x = Age, y = n, fill = Age)) +  
 labs(title = 'Age of Customers') +  
 theme(axis.text.x = element\_text(size = 10)) +  
 scale\_fill\_brewer(palette = 'Blues') +  
 theme(legend.position="none")  
print(customers\_age\_vis)



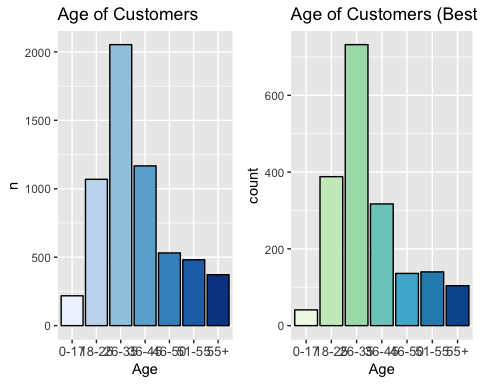
We can also plot a similar chart depicting the distribution of age within our “best seller” category. This will show us if there is a specific age category that purchased the best selling product more than other shoppers.

ageDist\_bs = ggplot(data = best\_seller) +  
 geom\_bar(color = 'black', mapping = aes(x = Age, y = ..count.., fill = Age)) +  
 labs(title = 'Age of Customers (Best Seller)') +  
 theme(axis.text.x = element\_text(size = 10)) +  
 scale\_fill\_brewer(palette = 'GnBu') +   
 theme(legend.position="none")  
print(ageDist\_bs)



It seems as though younger people (18-25 & 26-35) account for the highest number of purchases of the best selling product. Lets compare this observation to the overall dataset.

grid.arrange(customers\_age\_vis, ageDist\_bs, ncol=2)



We can see that there is some deviation with the proportion of customers grouped by age when comparing the best selling product to the overall dataset. It looks like older customers > Age 45 are buying the top seller slightly less than other products included in the overall dataset.

Now that we have examined age, lets move to another variable.