E-commerce Shipping Data

Zaryn Ooi

8/5/2021

Backgroud of the Company

An international e-commerce company based wants to discover key insights from their customer database. They want to use some of the most advanced machine learning techniques to study their customers. The company sells electronic products.

Task

1) Relation Between Customer Rating vs Product Arrived on Time
2) Relation Between Customer Care Calls vs Cost of The Product
3) Discount Offered over Product Importance
4) Which Mode of Shipment is the Most Desirable?
5) Which Gender Made the Most Prior Purchase?

Data Source:

E-Commerce Shipping Data (Version 1)

https://www.kaggle.com/prachi13/customer-analytics/code

This dataset is used for model building contained 10999 observations of 12 variables. The data contains the following information:

- ID: ID Number of customers.
- Warehouse block: The company have big warehouse which is divided in to block such as A,B,C,D,E.
- Mode of shipment:The company ships the products in multiple way such as Ship, Flight and Road.
- Customer care calls: The number of calls made from enquiry for enquiry of the shipment.
- Customer rating: The company has rated from every customer. 1 is the lowest (Worst), 5 is the highest (Best).
- Cost of the product: Cost of the product (USD).
- Prior purchases: Number of prior purchase.
- Product importance: The company has categorized the importance of product in various parameter such as low, medium, high.
- Gender: Male and Female.
- Discount offered: Discount offered on that specific product.
- Weight in gms: Product weight in grams.
- Reached on time: It is the target variable,where 1 indicates that product has NOT reached on time and 0 indicates that products has reached on time.

Install and Load Packages

```
install.packages('tidyverse')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages('dplyr')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages('ggplot2')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages('lubridate')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages('skimr')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(skimr)
```

```
install.packages('caret')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages('rpart')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```
install.packages('randomForest')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
install.packages('matrixStats')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
install.packages('gbm')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
install.packages('data.table')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
install.packages('gridExtra')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
install.packages('corrplot')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
install.packages('rpart.plot')
```

```
## Installing package into '/home/rstudio-user/R/x86_64-pc-linux-gnu-library/4.1'
## (as 'lib' is unspecified)
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(rpart)
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
library(matrixStats)
```

```
##
```

```
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count
```

```r
library(gbm)
```

```
## Loaded gbm 2.1.8
```

```r
library(data.table)
```

```
##
## Attaching package: 'data.table'

## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year

## The following objects are masked from 'package:dplyr':
##
##     between, first, last

## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

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

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

```r
library(corrplot)
```

```
## corrplot 0.90 loaded
```

```r
library(rpart.plot)
```

Importing Data

```r
Train <- read_csv("Train.csv")
```

```
## Rows: 10999 Columns: 12

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Warehouse_block, Mode_of_Shipment, Product_importance, Gender
## dbl (8): ID, Customer_care_calls, Customer_rating, Cost_of_the_Product, Prio...

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Explore the Imported Data

Let's use the `head()` function and `skim_without_charts()` function to display the data.

```
head(Train)
```

```
## # A tibble: 6 x 12
##      ID Warehouse_block Mode_of_Shipment Customer_care_calls Customer_rating
##   <dbl> <chr>           <chr>                          <dbl>           <dbl>
## 1     1 D               Flight                             4               2
## 2     2 F               Flight                             4               5
## 3     3 A               Flight                             2               2
## 4     4 B               Flight                             3               3
## 5     5 C               Flight                             2               2
## 6     6 F               Flight                             3               1
## # ... with 7 more variables: Cost_of_the_Product <dbl>, Prior_purchases <dbl>,
## #   Product_importance <chr>, Gender <chr>, Discount_offered <dbl>,
## #   Weight_in_gms <dbl>, Reached.on.Time_Y.N <dbl>
```

```
skim_without_charts(Train)
```

Table 1: Data summary

Name	Train
Number of rows	10999
Number of columns	12
Column type frequency:	
character	4
numeric	8
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
Warehouse_block	0	1	1	1	0	5	0
Mode_of_Shipment	0	1	4	6	0	3	0
Product_importance	0	1	3	6	0	3	0
Gender	0	1	1	1	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
ID	0	1	5500.00	3175.28	1	2750.5	5500	8249.5	10999
Customer_care_calls	0	1	4.05	1.14	2	3.0	4	5.0	7
Customer_rating	0	1	2.99	1.41	1	2.0	3	4.0	5
Cost_of_the_Product	0	1	210.20	48.06	96	169.0	214	251.0	310
Prior_purchases	0	1	3.57	1.52	2	3.0	3	4.0	10
Discount_offered	0	1	13.37	16.21	1	4.0	7	10.0	65
Weight_in_gms	0	1	3634.02	1635.38	1001	1839.5	4149	5050.0	7846

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
Reached.on.Time_Y.N	0	1	0.60	0.49	0	0.0	1	1.0	1

Now, let's check whether the 'Train' data is a NULL object with the `is.null()` function.

```
is.null(Train)
```

```
## [1] FALSE
```

The `is.null()` function returns FALSE, which means that the data drame is not a NULL object.

Let's see if there's any duplicates in the 'Train' data frame.

```
sum(duplicated(Train))
```

```
## [1] 0
```

There is no duplicated data in the data frame. We can now dove into the analysing the data.

Summary Statistics of 'Train' Data:

```
Train %>%
  select(Warehouse_block,
         Mode_of_Shipment,
         Customer_care_calls,
         Customer_rating,
         Cost_of_the_Product,
         Prior_purchases,
         Gender,
         Discount_offered) %>%
  summary()
```

```
##  Warehouse_block    Mode_of_Shipment   Customer_care_calls Customer_rating
##  Length:10999       Length:10999       Min.   :2.000       Min.   :1.000  
##  Class :character   Class :character   1st Qu.:3.000       1st Qu.:2.000  
##  Mode  :character   Mode  :character   Median :4.000       Median :3.000  
##                                        Mean   :4.054       Mean   :2.991  
##                                        3rd Qu.:5.000       3rd Qu.:4.000  
##                                        Max.   :7.000       Max.   :5.000  
##  Cost_of_the_Product Prior_purchases     Gender          Discount_offered
##  Min.   : 96.0       Min.   : 2.000   Length:10999       Min.   : 1.00   
##  1st Qu.:169.0       1st Qu.: 3.000   Class :character   1st Qu.: 4.00   
##  Median :214.0       Median : 3.000   Mode  :character   Median : 7.00   
##  Mean   :210.2       Mean   : 3.568                      Mean   :13.37   
##  3rd Qu.:251.0       3rd Qu.: 4.000                      3rd Qu.:10.00   
##  Max.   :310.0       Max.   :10.000                      Max.   :65.00   
```

Exploratory Visualizations

I'll first rename the 'Reached.on.Time_Y.N' column to 'Reached_On_Time' to make it easier when plotting the visualizations.

```
Train <-
  Train %>%
rename(Reached_On_Time = Reached.on.Time_Y.N)
```

Then, I used the `mutate()` function to create a new column ‘Received_On_Time’ to modify the ‘Reached_On_Time’ column, where 1 is No and 0 is Yes.

```
mutate(Train, Received_On_Time = ifelse(Train$Reached_On_Time == "0","Yes","No"))
```

```
## # A tibble: 10,999 x 13
##       ID Warehouse_block Mode_of_Shipment Customer_care_calls Customer_rating
##    <dbl> <chr>           <chr>                          <dbl>           <dbl>
##  1     1 D               Flight                             4               2
##  2     2 F               Flight                             4               5
##  3     3 A               Flight                             2               2
##  4     4 B               Flight                             3               3
##  5     5 C               Flight                             2               2
##  6     6 F               Flight                             3               1
##  7     7 D               Flight                             3               4
##  8     8 F               Flight                             4               1
##  9     9 A               Flight                             3               4
## 10    10 B               Flight                             3               2
## # ... with 10,989 more rows, and 8 more variables: Cost_of_the_Product <dbl>,
## #   Prior_purchases <dbl>, Product_importance <chr>, Gender <chr>,
## #   Discount_offered <dbl>, Weight_in_gms <dbl>, Reached_On_Time <dbl>,
## #   Received_On_Time <chr>
```

```
Train <- Train %>%
  mutate(Train, Received_On_Time = ifelse(Train$Reached_On_Time == "0","Yes","No"))
```

1) Relation Between Customer Rating over Product Arrived on Time

I would like to identify if the product will reached on time for customers who gave the company good ratings than those who gave poor ratings. In this case, 1 is the lowest, while 5 is the highest. Hence, I will create a new summarized table where I will classify the Customer Rating into “Good Rating” and “Poor Rating” for the analysis.

```
min(Train$Customer_rating)
```

```
## [1] 1
```

```
max(Train$Customer_rating)
```

```
## [1] 5
```

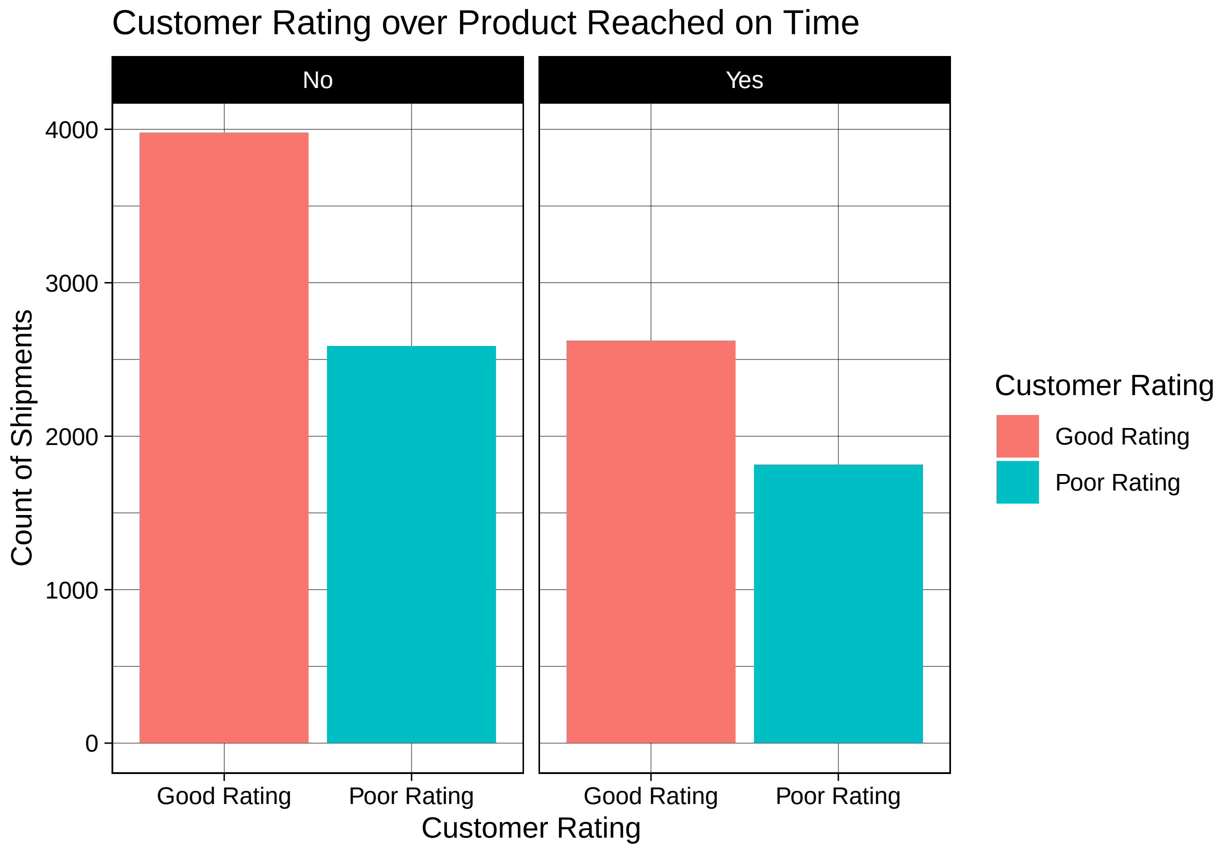
```
Rating <-
  Train %>%
   summarize(
    Customer_Rating = factor(case_when(
      Customer_rating >= 3 ~ "Good Rating",
      Customer_rating <3 ~ "Poor Rating",
      ), levels = c("Good Rating", "Poor Rating")),
      Received_On_Time, .groups="drop") %>%
      drop_na()

head(Rating)
```

```
## # A tibble: 6 x 2
##   Customer_Rating Received_On_Time
##   <fct>           <chr>
## 1 Poor Rating     No
## 2 Good Rating     No
## 3 Poor Rating      No
## 4 Good Rating      No
## 5 Poor Rating      No
## 6 Poor Rating      No
```

Visualization I will create a barchart that represents the Product Reached on Time based on the two categories of Customer Rating. It is faceted by two categories, where 'No' indicates Product Did Not Reached on Time and 'Yes' indicates Product Reached on Time.

```
ggplot(data = Rating)+
  geom_bar(mapping= aes(x= Customer_Rating, fill= Customer_Rating)) +
  facet_wrap(~ Received_On_Time) +
  theme_linedraw()+
  xlab ("Customer Rating") +
  ylab ("Count of Shipments") +
  scale_fill_discrete(name = "Customer Rating", labels = c("Good Rating", "Poor Rating")) +
labs(title = "Customer Rating over Product Reached on Time")
```



Analysis:

From the bar chart, we can identify that most product did not reached on time. This barchart also shows that most customers who received their product on time are customers who gave good ratings. However, most customers who did not received their product on time are also customers who gave good ratings. Hence, this reveals that there is no correlation between customer ratings and product reached on time.

2) Relation Between Customer Care Calls over Cost of The Product

To identify the relationship between Customer Care Calls and Cost of Product, I will create a new summarized table where I will classify the cost of the product into more easily interpretable categories for the analysis.

```
min(Train$Cost_of_the_Product)
```

```
## [1] 96
```

```
max(Train$Cost_of_the_Product)
```

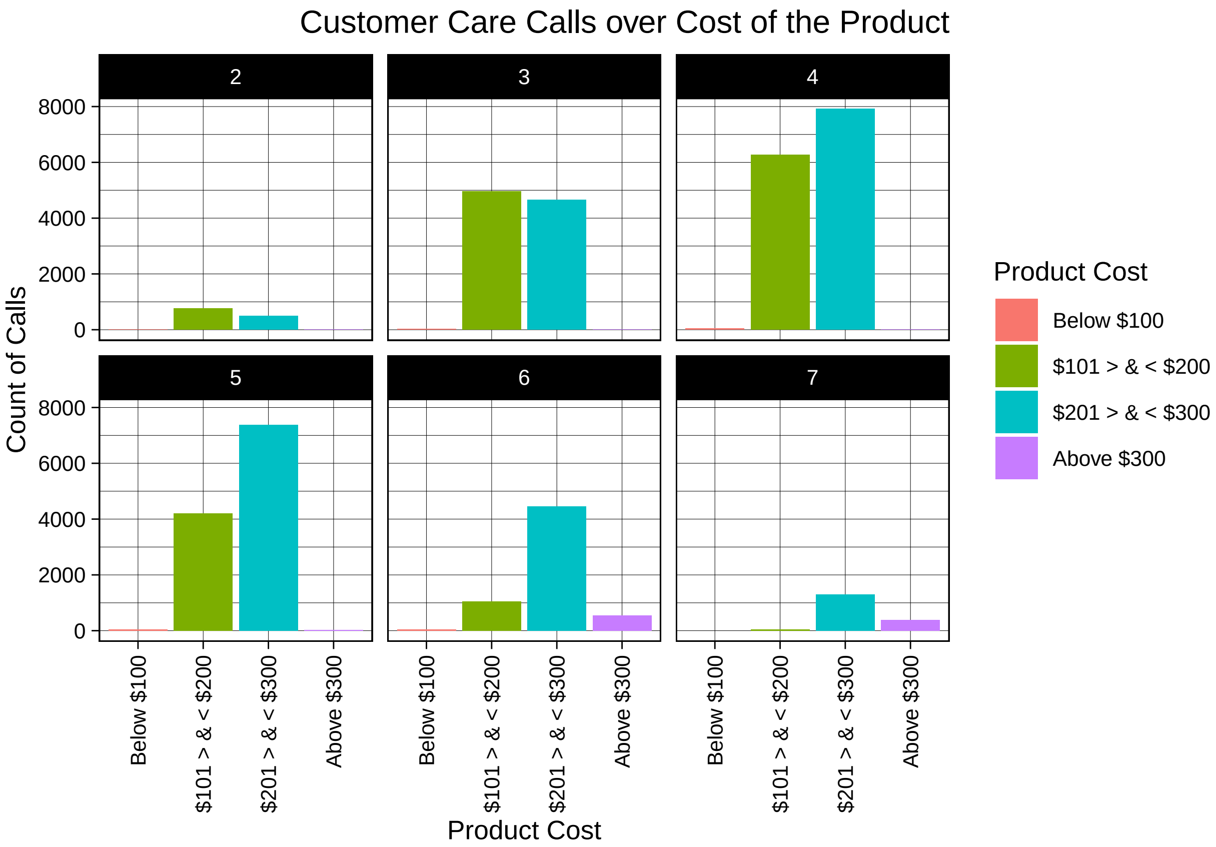
```
## [1] 310
```

```
Cost <- Train %>%
   summarise(
    Cost_of_the_Product = factor(case_when(
      Cost_of_the_Product >= 96 & Cost_of_the_Product <= 100 ~ "Below $100",
      Cost_of_the_Product >= 101 & Cost_of_the_Product <= 200 ~ "$101 > & < $200",
      Cost_of_the_Product >= 201 & Cost_of_the_Product <= 300 ~ "$201 > & < $300",
      Cost_of_the_Product >= 301 & Cost_of_the_Product <= 310 ~ "Above $300",
      ), levels = c("Below $100", "$101 > & < $200", "$201 > & < $300", "Above $300")),
      Customer_care_calls, .groups="drop") %>%
      drop_na()
head(Cost)
```

```
## # A tibble: 6 x 2
##   Cost_of_the_Product Customer_care_calls
##   <fct>                             <dbl>
## 1 $101 > & < $200                       4
## 2 $201 > & < $300                       4
## 3 $101 > & < $200                       2
## 4 $101 > & < $200                       3
## 5 $101 > & < $200                       2
## 6 $101 > & < $200                       3
```

Visualization For the visualization, I will create a barchart that represents the customer care calls based on the cost of the product and then it is faceted by the four categories of cost of the product. By doing this, I'm able to identify the relationship between the customer care calls and the cost of the product.

```
ggplot(data = Cost,aes(Cost_of_the_Product,Customer_care_calls,fill = Cost_of_the_Product)) +
    geom_col() +
    facet_wrap(~Customer_care_calls)+
    theme_linedraw()+
    xlab ("Product Cost") +
    ylab ("Count of Calls") +
    scale_fill_discrete(name = "Product Cost", labels = c("Below $100", "$101 > & < $200", "$201 > & < $
    labs(title="Customer Care Calls over Cost of the Product") +
    theme(legend.position="right", text = element_text(size = 10), plot.title = element_text(hjust = 1)
```



Analysis:

Here, we can see customer who bought products that costs below 100 did not made calls to make enquiry for the shipment. As seen above, customers who urchased products that costs between 100 to 300 start to made 2 to 7 calls to make enquiry for the shipment. In this case, most of these customers made 4 calls to ask for their shipment.

Additionally, the prices of the products in 6 or 7 calls to customer increases to above 300. In short, as the cost of the product increases, the number of calls made by customer for shipment enquiry will also increase. Reason being is that they might worry that they will lost shipment that worth more than $300.

3) Discount Offered over Product Importance

I will create a new summarized table named "Discount" where I classified the Discount Offered into more easily interpretable categories for the analysis.

```
Discount <- Train %>%
   summarise(
    Discount_Offered = factor(case_when(
      Discount_offered <= 9  ~ "Min $10 off",
      Discount_offered >= 10 & Discount_offered <= 19 ~ "$10 to $19 off",
      Discount_offered >= 20 & Discount_offered <= 29 ~ "$20 to $29 off",
      Discount_offered >= 30 & Discount_offered <= 39 ~ "$30 to $39 off",
      Discount_offered >= 40 & Discount_offered <= 49 ~ "$40 to $49 off",
      Discount_offered >= 50 & Discount_offered <= 59 ~ "$50 t0 $59 off",
      Discount_offered >= 60 ~ "Up to $60 off"
      ), levels = c("Min $10 off", "$10 to $19 off", "$20 to $29 off", "$30 to $39 off","$40 to $49 off
      Product_importance, .groups="drop") %>%
      drop_na()
```

```
dsct <- table(Discount$Discount_Offered,Discount$Product_importance)
dsct
```

```
## 
##                   high  low medium
##   Min $10 off      613 3638   3241
##   $10 to $19 off   105  636    557
##   $20 to $29 off    55  219    191
##   $30 to $39 off    47  215    202
##   $40 to $49 off    49  242    216
##   $50 to $59 off     0    0      0
##   Up to $60 off     34  125    128
```

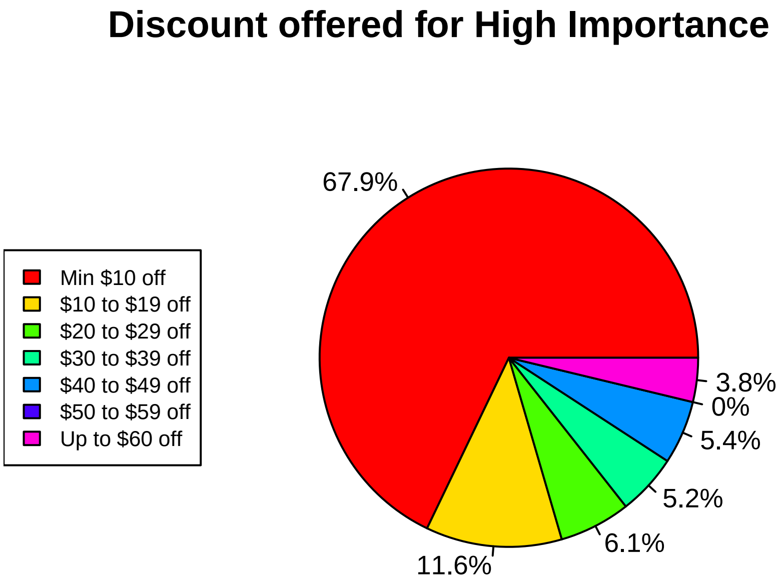
```
dsct <- as.data.frame(dsct)
```

Visualization I will create three pie charts where I can identify the percentage of discount offered over three types of product importance. By doing this, I can find out the largest category of discount offered based on the importance of the product.

```
discount_high <- c(613, 105, 55, 47, 49, 0, 34)
labels <- c("Min $10 off", "$10 to $19 off", "$20 to $29 off", "$30 to $39 off","$40 to $49 off","$50 to

piepercent_high <- paste0(round( discount_high / sum(discount_high) * 100, 1), "%")

pie(discount_high, piepercent_high, cex = 0.8,
main =" Discount offered for High Importance Product",
      col = rainbow(length(discount_high)))
      legend("left", c("Min $10 off", "$10 to $19 off", "$20 to $29 off", "$30 to $39 off","$40 to $49 o
      fill = rainbow(length(discount_high))
      )
```



Analysis:

The top 3 categories of Discount Offered for "High" importance product is :

1) Min $10 off (67.9%)

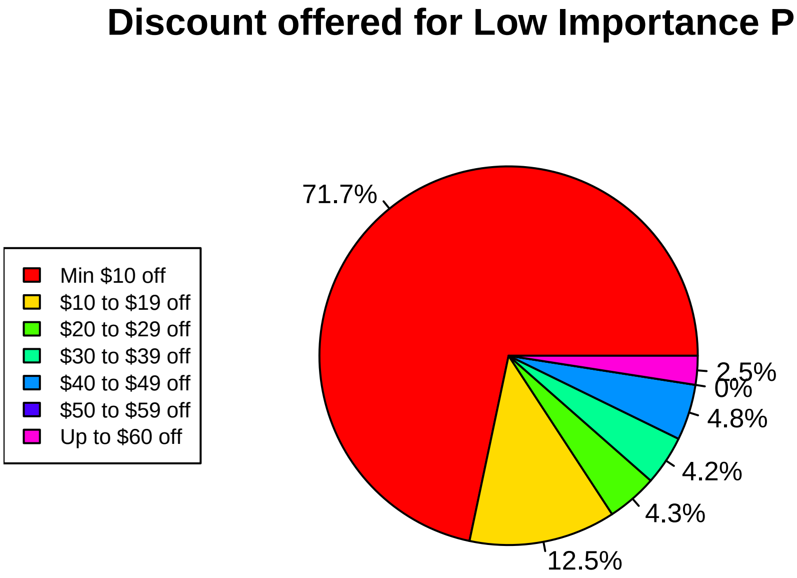
2) $10 to $19 off (11.6%)
3) $20 to $29 off (6.1%)

Besides, the pie chart also shows that there is no seller who offered $50 to $59 off (0%) for “High” importance product.

```
discount_low <- c(3638, 636, 219, 215, 242, 0, 125)
labels <- c("Min $10 off", "$10 to $19 off", "$20 to $29 off", "$30 to $39 off","$40 to $49 off","$50 t

piepercent_low <- paste0(round( discount_low / sum(discount_low) * 100, 1), "%")

pie(discount_low, piepercent_low, cex = 0.8,
main ="Discount offered for Low Importance Product",
      col = rainbow(length(discount_low)))
      legend("left", c("Min $10 off", "$10 to $19 off", "$20 to $29 off", "$30 to $39 off","$40 to $49
      fill = rainbow(length(discount_low))
      )
```



Analysis:

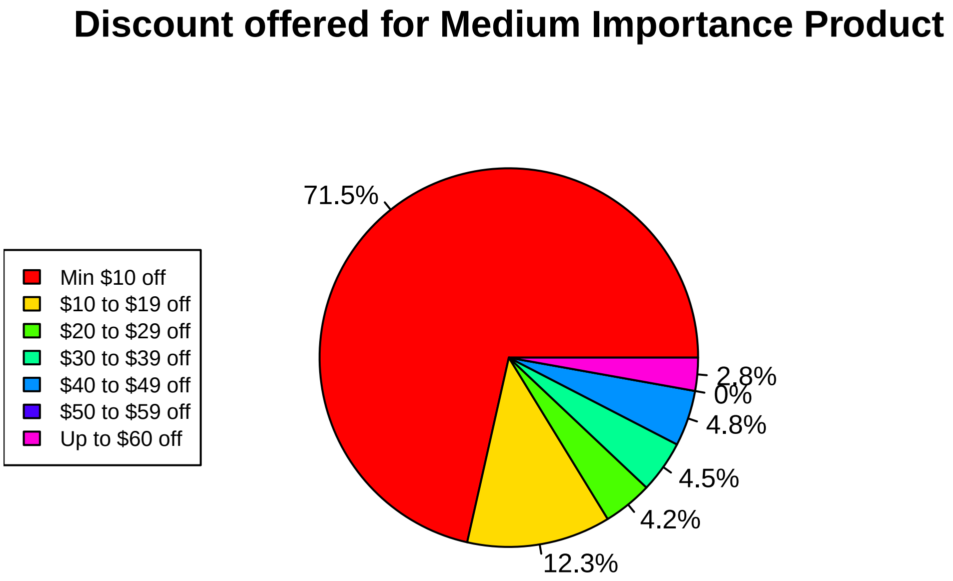
The top 3 categories of Discount Offered for “Low” importance product is : 1) Min $10 off (71.7%) 2) $10 to $19 off (12.5%) 3) $40 to $49 off (4.8%)

Similarly, there is no seller who offered $50 to $59 off (0%) for “Low” importance product.

```
discount_medium <- c(3241, 557, 191, 202, 216, 0, 128)
labels <- c("Min $10 off", "$10 to $19 off", "$20 to $29 off", "$30 to $39 off","$40 to $49 off","$50 t

piepercent_medium <- paste0(round( discount_medium / sum(discount_medium) * 100, 1), "%")

pie(discount_medium, piepercent_medium, cex = 0.8,
main ="Discount offered for Medium Importance Product",
      col = rainbow(length(discount_medium)))
      legend("left", c("Min $10 off", "$10 to $19 off", "$20 to $29 off", "$30 to $39 off","$40 to $49
      fill = rainbow(length(discount_medium))
      )
```



Analysis:

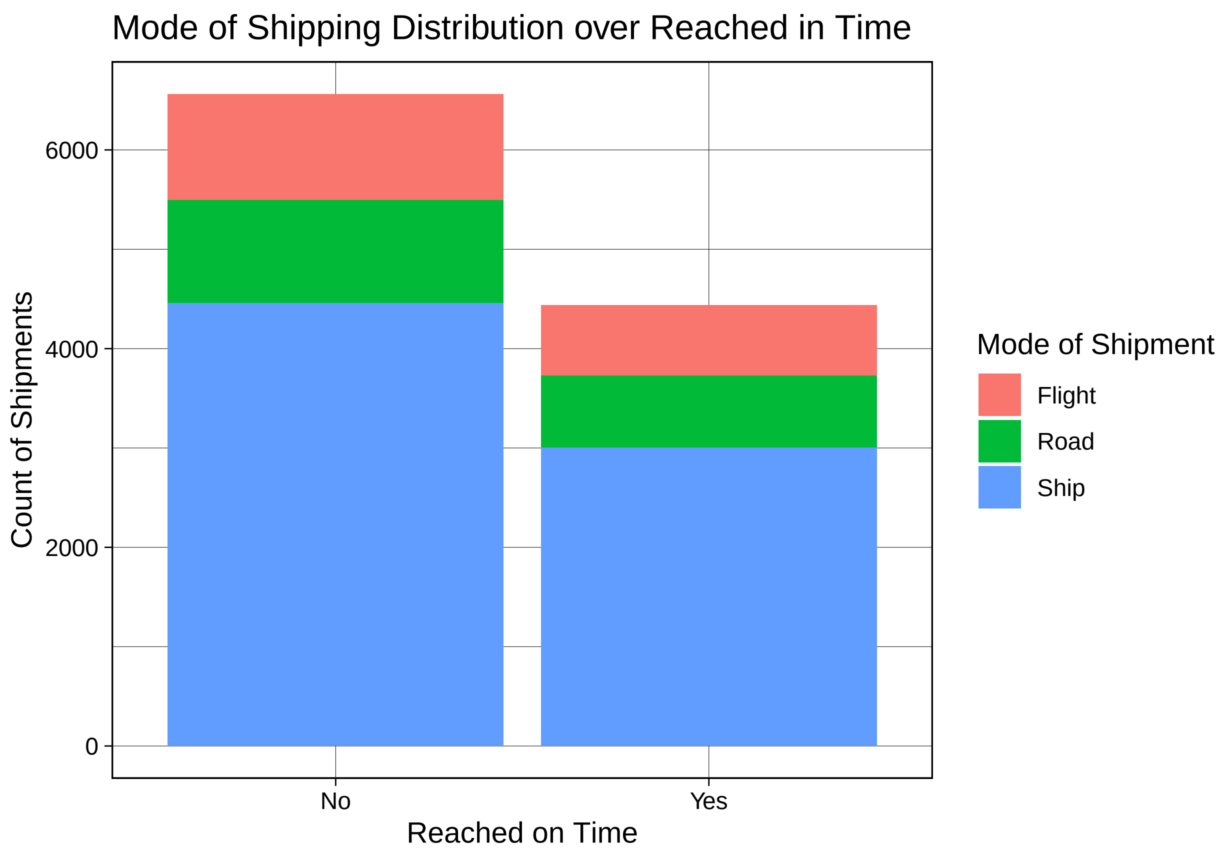
The top 3 categories of Discount Offered for "Medium" importance product is : 1) Min $10 off - 71.5% 2) $10 to $19 off - 12.3% 3) $40 to $49 off - 4.8%

Just like "High" and "Low" importance product, sellers did not offer $50 to $59 off(0%) for "Medium" importance product.

4) Which Mode of Shipments is the most desirable?

Visualization I will create a barchart that represents the Product Reached On Time based on Mode of Shipment.

```
ggplot(data = Train, aes(Received_On_Time)) +
  geom_bar(aes(fill = Mode_of_Shipment)) +
  theme_linedraw() +
  xlab ("Reached on Time") +
  ylab ("Count of Shipments") +
  scale_fill_discrete(name = "Mode of Shipment", labels = c("Flight", "Road", "Ship")) + labs(title = "
```



Analysis:

According to the barchart above, we can see that most sellers prefer to ship their products to customers via ship shipment. Besides, by comparing whether the product has reached on time, it seems that there are more shipments did not reached on time that shipments that reached on time. For shipment that reached on time, mostly were transported by ship. Hence, Ship appears to be the fastest and most desirable mode of transportation.

5) Which Gender Made the Most Prior Purchase?

I will create a new table, "PriorP" for easier analysis.

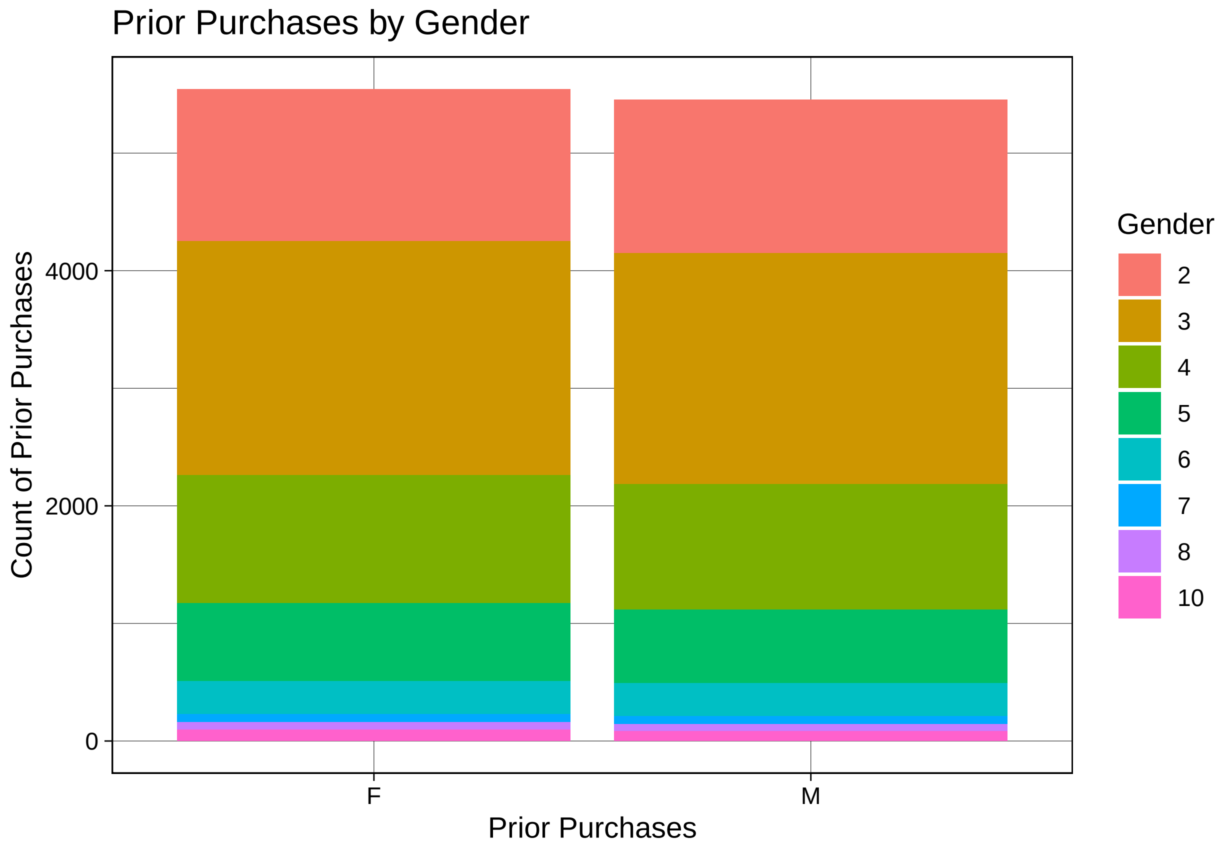
```
PriorP <- table(Train$Gender,Train$Prior_purchases)
PriorP
```

```
##
##        2    3    4    5    6    7    8   10
##   F 1295 1989 1089  662  280   70   66   94
##   M 1304 1966 1066  625  281   66   62   84
```

```
PriorP <- as.data.frame(PriorP)
```

Visualization I will create a barchart that represents the Prior Purchases based on Gender.

```
ggplot(data = PriorP, aes(Var1, Freq)) +
  geom_col(aes(fill = Var2)) +
  theme_linedraw() +
  xlab ("Prior Purchases") +
  ylab ("Count of Prior Purchases") +
scale_fill_discrete(name = "Gender", labels = c("2","3","4","5","6","7","8","10")) + labs(title = "Pr
```



Analysis:

According to the graph, Female customer had more prior purchase than Male customer, but the prior purchases of both genders are almost the same.

git config –global user.email “you@example.com” git config –global user.name “Your Name”