BA64036\_Assignment2

Durga Prasad Gandi

2023-10-14

## install required library

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

# load the retailset

retail = read.csv("C:/Users/gdurg/Desktop/rhistory/Online\_Retail.csv")  
  
head(retail)

## InvoiceNo StockCode Description Quantity  
## 1 536365 85123A WHITE HANGING HEART T-LIGHT HOLDER 6  
## 2 536365 71053 WHITE METAL LANTERN 6  
## 3 536365 84406B CREAM CUPID HEARTS COAT HANGER 8  
## 4 536365 84029G KNITTED UNION FLAG HOT WATER BOTTLE 6  
## 5 536365 84029E RED WOOLLY HOTTIE WHITE HEART. 6  
## 6 536365 22752 SET 7 BABUSHKA NESTING BOXES 2  
## InvoiceDate UnitPrice CustomerID Country  
## 1 12/1/2010 8:26 2.55 17850 United Kingdom  
## 2 12/1/2010 8:26 3.39 17850 United Kingdom  
## 3 12/1/2010 8:26 2.75 17850 United Kingdom  
## 4 12/1/2010 8:26 3.39 17850 United Kingdom  
## 5 12/1/2010 8:26 3.39 17850 United Kingdom  
## 6 12/1/2010 8:26 7.65 17850 United Kingdom

1. Show the breakdown of the number of transactions by countries i.e., how many transactions are in the retailset for each country (consider all records including cancelled transactions). Show this in total number and also in percentage. Show only countries accounting for more than 1% of the total transactions.

## Compute Number of transactions by Countries

country\_count = retail %>% group\_by(Country) %>% count(Country)  
  
country\_count #Number of transactions for each country

## # A tibble: 38 × 2  
## # Groups: Country [38]  
## Country n  
## <chr> <int>  
## 1 Australia 1259  
## 2 Austria 401  
## 3 Bahrain 19  
## 4 Belgium 2069  
## 5 Brazil 32  
## 6 Canada 151  
## 7 Channel Islands 758  
## 8 Cyprus 622  
## 9 Czech Republic 30  
## 10 Denmark 389  
## # ℹ 28 more rows

country\_percentage = retail %>% group\_by(Country) %>% summarise(percent = 100\* n()/nrow(retail))  
  
country\_percentage #This gives the percentage of transactiond for each Country

## # A tibble: 38 × 2  
## Country percent  
## <chr> <dbl>  
## 1 Australia 0.232   
## 2 Austria 0.0740   
## 3 Bahrain 0.00351  
## 4 Belgium 0.382   
## 5 Brazil 0.00591  
## 6 Canada 0.0279   
## 7 Channel Islands 0.140   
## 8 Cyprus 0.115   
## 9 Czech Republic 0.00554  
## 10 Denmark 0.0718   
## # ℹ 28 more rows

filter = filter(country\_percentage, percent>1) #which gives value greater than 1 in retailset  
  
filter

## # A tibble: 4 × 2  
## Country percent  
## <chr> <dbl>  
## 1 EIRE 1.51  
## 2 France 1.58  
## 3 Germany 1.75  
## 4 United Kingdom 91.4

1. Create a new variable ‘TransactionValue’ that is the product of the exising ‘Quantity’ and ‘UnitPrice’ variables. Add this variable to the retailframe.

retail$TransactionValue = retail$Quantity \* retail$UnitPrice  
  
head(retail$TransactionValue)

## [1] 15.30 20.34 22.00 20.34 20.34 15.30

# The data will be available by using head of retailset.

1. Using the newly created variable, TransactionValue, show the breakdown of transaction values by countries i.e. how much money in total has been spent each country. Show this in total sum of transaction values. Show only countries with total transaction exceeding 130,000 British Pound.

country\_transaction = retail %>% group\_by(Country) %>% summarise(sum=sum(TransactionValue))  
  
  
filtered\_country\_transaction = filter(country\_transaction, country\_transaction$sum>13000) #To get Transaction values greater than 13000   
  
filtered\_country\_transaction

## # A tibble: 17 × 2  
## Country sum  
## <chr> <dbl>  
## 1 Australia 137077.  
## 2 Belgium 40911.  
## 3 Channel Islands 20086.  
## 4 Denmark 18768.  
## 5 EIRE 263277.  
## 6 Finland 22327.  
## 7 France 197404.  
## 8 Germany 221698.  
## 9 Italy 16891.  
## 10 Japan 35341.  
## 11 Netherlands 284662.  
## 12 Norway 35163.  
## 13 Portugal 29367.  
## 14 Spain 54775.  
## 15 Sweden 36596.  
## 16 Switzerland 56385.  
## 17 United Kingdom 8187806.

1. This is an optional question which carries additional marks (golden questions). In this question, we are dealing with the InvoiceDate variable. The variable is read as a categorical when you read retail from the file. Now we need to explicitly instruct R to interpret this as a Date variable. “POSIXlt” and “POSIXct” are two powerful object classes in R to deal with date and time. Click here for more information. First let’s convert ‘InvoiceDate’ into a POSIXlt object: Temp=strptime(Online\_RetailNew\_Invoice\_Date <- as.Date(Temp) The Date objects have a lot of flexible functions. For example knowing two date values, the object allows you to know the difference between the two dates in terms of the number days. Try this: Online\_RetailNew\_Invoice\_Date[10] Also we can convert dates to days of the week. Let’s define a new variable for that Online\_RetailNew\_Invoice\_Date) For the Hour, let’s just take the hour (ignore the minute) and convert into a normal numerical value: Online\_Retail$New\_Invoice\_Hour = as.numeric(format(Temp, "%H")) Finally, lets define the month as a separate numeric variable too: Page 3 Online\_Retail$New\_Invoice\_Month = as.numeric(format(Temp, “%m”))

Temp = strptime(retail$InvoiceDate, format = '%m/%d/%Y %H:%M', tz = 'GMT')  
head(Temp)

## [1] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"  
## [3] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"  
## [5] "2010-12-01 08:26:00 GMT" "2010-12-01 08:26:00 GMT"

retail$New\_Invoice\_Date = as.Date(Temp)  
retail$Invoice\_Day\_Week = weekdays(retail$New\_Invoice\_Date)  
retail$New\_Invoice\_Hour = as.numeric(format(Temp, "%H"))  
retail$New\_Invoice\_Month = as.numeric(format(Temp, "%m"))

head(retail)

## InvoiceNo StockCode Description Quantity  
## 1 536365 85123A WHITE HANGING HEART T-LIGHT HOLDER 6  
## 2 536365 71053 WHITE METAL LANTERN 6  
## 3 536365 84406B CREAM CUPID HEARTS COAT HANGER 8  
## 4 536365 84029G KNITTED UNION FLAG HOT WATER BOTTLE 6  
## 5 536365 84029E RED WOOLLY HOTTIE WHITE HEART. 6  
## 6 536365 22752 SET 7 BABUSHKA NESTING BOXES 2  
## InvoiceDate UnitPrice CustomerID Country TransactionValue  
## 1 12/1/2010 8:26 2.55 17850 United Kingdom 15.30  
## 2 12/1/2010 8:26 3.39 17850 United Kingdom 20.34  
## 3 12/1/2010 8:26 2.75 17850 United Kingdom 22.00  
## 4 12/1/2010 8:26 3.39 17850 United Kingdom 20.34  
## 5 12/1/2010 8:26 3.39 17850 United Kingdom 20.34  
## 6 12/1/2010 8:26 7.65 17850 United Kingdom 15.30  
## New\_Invoice\_Date Invoice\_Day\_Week New\_Invoice\_Hour New\_Invoice\_Month  
## 1 2010-12-01 Wednesday 8 12  
## 2 2010-12-01 Wednesday 8 12  
## 3 2010-12-01 Wednesday 8 12  
## 4 2010-12-01 Wednesday 8 12  
## 5 2010-12-01 Wednesday 8 12  
## 6 2010-12-01 Wednesday 8 12

1. Show the percentage of transactions (by numbers) by days of the week

day\_per = retail %>% group\_by(Invoice\_Day\_Week) %>% summarise(percent = 100\* n()/nrow(retail))  
  
day\_per

## # A tibble: 6 × 2  
## Invoice\_Day\_Week percent  
## <chr> <dbl>  
## 1 Friday 15.2  
## 2 Monday 17.6  
## 3 Sunday 11.9  
## 4 Thursday 19.2  
## 5 Tuesday 18.8  
## 6 Wednesday 17.5

1. Show the percentage of transactions (by transaction volume) by days of the week.

day\_transaction = retail %>% group\_by(Invoice\_Day\_Week) %>% summarise(sum= sum(TransactionValue))  
  
day\_transaction\_per = 100\*(day\_transaction$sum)/sum(day\_transaction$sum)  
  
day\_transaction$sum = day\_transaction\_per  
  
day\_transaction

## # A tibble: 6 × 2  
## Invoice\_Day\_Week sum  
## <chr> <dbl>  
## 1 Friday 15.8   
## 2 Monday 16.3   
## 3 Sunday 8.27  
## 4 Thursday 21.7   
## 5 Tuesday 20.2   
## 6 Wednesday 17.8

1. Show the percentage of transactions (by transaction volume) by month of the year.

month\_transaction = retail %>% group\_by(New\_Invoice\_Month) %>% summarise(sum= sum(TransactionValue))  
  
month\_transaction\_per = 100\*(month\_transaction$sum)/sum(month\_transaction$sum)  
  
month\_transaction$sum = month\_transaction\_per  
  
month\_transaction

## # A tibble: 12 × 2  
## New\_Invoice\_Month sum  
## <dbl> <dbl>  
## 1 1 5.74  
## 2 2 5.11  
## 3 3 7.01  
## 4 4 5.06  
## 5 5 7.42  
## 6 6 7.09  
## 7 7 6.99  
## 8 8 7.00  
## 9 9 10.5   
## 10 10 11.0   
## 11 11 15.0   
## 12 12 12.1

1. What was the date with the highest number of transactions from Australia?

australia = retail %>%   
filter(Country=='Australia')%>% group\_by(New\_Invoice\_Date)%>%  
summarise(Number=sum(Quantity),amount=sum(TransactionValue))%>%  
arrange(desc(Number)) #desc used to arrange the values in descending order  
  
date\_highest\_transactions = australia$New\_Invoice\_Date[which.max(table(australia$New\_Invoice\_Date))]  
#To calculate maximum value which.max is used here.  
  
  
date\_highest\_transactions

## [1] "2011-06-15"

***The date for maximum transaction is 2011-06-15.***

1. The company needs to shut down the website for two consecutive hours for maintenance. What would be the hour of the day to start this so that the distribution is at minimum for the customers? The responsible IT team is available from 7:00 to 20:00 every day.

hour\_transaction\_retail = retail %>% group\_by(New\_Invoice\_Hour) %>% summarize(Count = n()) %>% filter(New\_Invoice\_Hour >= 7 & New\_Invoice\_Hour <= 20) %>% arrange(Count)  
  
optimal\_start\_hour = hour\_transaction\_retail %>% slice(1) %>% pull(New\_Invoice\_Hour)  
  
optimal\_start\_hour2 = ifelse(optimal\_start\_hour == 20, optimal\_start\_hour - 1, optimal\_start\_hour + 1)  
  
start\_of\_maintenance = paste(optimal\_start\_hour, ":00", sep = "")  
end\_of\_maintenance = paste(optimal\_start\_hour2, ":00", sep = "")  
  
start\_of\_maintenance

## [1] "7:00"

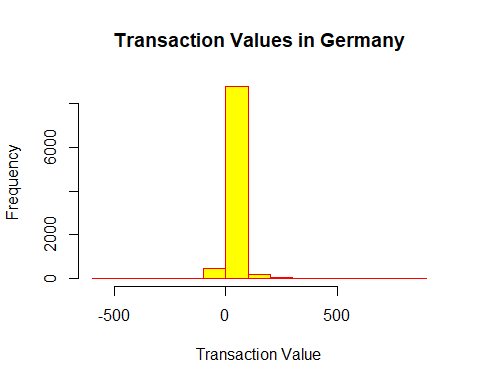
end\_of\_maintenance

## [1] "8:00"

***7,8 are two consecutive hours for maintenance.***

1. Plot the histogram of transaction values from Germany. Use the hist() function to plot.

germany = retail[retail$Country == "Germany", ]  
  
hist(germany$TransactionValue, main = "Transaction Values in Germany", xlab = "Transaction Value" , ylab = "Frequency",col = "yellow", border = "red")

 6. Which customer had the highest number of transactions? Which customer is most valuable (i.e. highest total sum of transactions)?

customer\_highest\_transaction =retail %>% group\_by(CustomerID)%>%  
summarise(CustomerTransaction = n())%>% filter(CustomerID != "NA")%>% filter(CustomerTransaction ==max(CustomerTransaction) )  
  
customer\_highest\_transaction$CustomerID

## [1] 17841

customer\_transaction\_total = retail %>% group\_by(CustomerID)%>%  
summarise(total.transaction.by.each.customer = sum(TransactionValue))%>% arrange(desc(total.transaction.by.each.customer))%>%  
filter(CustomerID != "NA")%>% filter(total.transaction.by.each.customer ==max(total.transaction.by.each.customer) )  
  
customer\_transaction\_total$CustomerID

## [1] 14646

***The maximum number of transactions is done by Customer Id:17841***

1. Calculate the percentage of missing values for each variable in the retailset (5% of total points). Hint colMeans():

missing\_percentage = colMeans(is.na(retail)) \* 100  
  
missing\_percentage

## InvoiceNo StockCode Description Quantity   
## 0.00000 0.00000 0.00000 0.00000   
## InvoiceDate UnitPrice CustomerID Country   
## 0.00000 0.00000 24.92669 0.00000   
## TransactionValue New\_Invoice\_Date Invoice\_Day\_Week New\_Invoice\_Hour   
## 0.00000 0.00000 0.00000 0.00000   
## New\_Invoice\_Month   
## 0.00000

***The customerId has more percentage of missing values for the above question***

1. What are the number of transactions with missing CustomerID records by countries?

missing\_customer\_transaction = retail[is.na(retail$CustomerID), ]  
missing\_customer\_counts\_country = table(missing\_customer\_transaction$Country)  
  
missing\_customer\_counts\_country

##   
## Bahrain EIRE France Hong Kong Israel   
## 2 711 66 288 47   
## Portugal Switzerland United Kingdom Unspecified   
## 39 125 133600 202

***The above data shows where Bahrain have least value(2) and UK has highest value (133600)in the retailset.***

1. On average, how often the costumers comeback to the website for their next shopping? (i.e. what is the average number of days between consecutive shopping) (5% of total points!) Hint: 1. A close approximation is also acceptable and you may find diff() function useful.

time\_diff = retail %>%   
 group\_by(CustomerID) %>%  
 mutate(difference.in.consecutivedays = c(0, diff(New\_Invoice\_Date))) %>%  
 filter(difference.in.consecutivedays > 0) %>%  
 ungroup()  
  
average\_time\_diff = mean(time\_diff$difference.in.consecutivedays)  
  
average\_time\_diff

## [1] 38.4875

***The average time difference between consecutive days is 38.4***

1. n the retail sector, it is very important to understand the return rate of the goods purchased by customers. In this example, we can define this quantity, simply, as the ratio of the number of transactions cancelled (regardless of the transaction value) over the total number of transactions. page 4 with this definition, what is the return rate for the French customers? (10% of total points). Consider the cancelled transactions as those where the ‘Quantity’ variable has a negative value.

french\_retail = filter(retail, Country == "France" )  
  
return\_rate = nrow(filter(french\_retail, Quantity<1)) / nrow(french\_retail)  
  
return\_rate

## [1] 0.01741264

***The return rate for the french retailset is 1.74%***

1. What is the product that has generated the highest revenue for the retailer? (i.e. item with the highest total sum of ‘TransactionValue’).

product\_revenue = retail %>% group\_by(StockCode) %>% summarise(sum = sum(TransactionValue))  
  
product\_revenue[which.max(product\_revenue$sum), ]

## # A tibble: 1 × 2  
## StockCode sum  
## <chr> <dbl>  
## 1 DOT 206245.

***The product with stockcode “DOT” is the one that has produced highest revenue.***

1. How many unique customers are represented in the retailset? You can use unique() and length() functions.

unique\_customers = length(unique(retail$CustomerID))  
  
unique\_customers

## [1] 4373

***we have total 4373 Unique customers in the retail.***