Assignment 2

keerthi Tiyyagura

2023-10-08

#Import the dataset "Online Retail"  
ORetail <- read.csv("C:/Users/keert/Downloads/Online\_Retail.csv")

1. Show the breakdown of the number of transactions by countries i.e., how many transactions are in the dataset 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.

#Showing the total number of transactions by country  
Country\_totalnumber <- table(ORetail$Country)  
Country\_totalnumber

##   
## Australia Austria Bahrain   
## 1259 401 19   
## Belgium Brazil Canada   
## 2069 32 151   
## Channel Islands Cyprus Czech Republic   
## 758 622 30   
## Denmark EIRE European Community   
## 389 8196 61   
## Finland France Germany   
## 695 8557 9495   
## Greece Hong Kong Iceland   
## 146 288 182   
## Israel Italy Japan   
## 297 803 358   
## Lebanon Lithuania Malta   
## 45 35 127   
## Netherlands Norway Poland   
## 2371 1086 341   
## Portugal RSA Saudi Arabia   
## 1519 58 10   
## Singapore Spain Sweden   
## 229 2533 462   
## Switzerland United Arab Emirates United Kingdom   
## 2002 68 495478   
## Unspecified USA   
## 446 291

# Calculate the percentage of transactions for each country  
transaction\_percent <- round(100\*prop.table(Country\_totalnumber),digits = 2)  
  
# Combine the total number and percentage of transactions into a table  
total <- data.frame(Country=names(Country\_totalnumber),  
 TotalNumber=Country\_totalnumber,  
 Percentage=transaction\_percent)  
  
# Subset the table to show only countries accounting for more than 1% of the total transactions  
total <- subset(total,transaction\_percent>1)  
total

## Country TotalNumber.Var1 TotalNumber.Freq Percentage.Var1  
## 11 EIRE EIRE 8196 EIRE  
## 14 France France 8557 France  
## 15 Germany Germany 9495 Germany  
## 36 United Kingdom United Kingdom 495478 United Kingdom  
## Percentage.Freq  
## 11 1.51  
## 14 1.58  
## 15 1.75  
## 36 91.43

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

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

#Creating a new variable ‘TransactionValue’  
ORetail <- ORetail %>% mutate(TransactionValue= Quantity \* UnitPrice)  
summary(ORetail$TransactionValue)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -168469.60 3.40 9.75 17.99 17.40 168469.60

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.

data <- summarise(group\_by(ORetail,Country),sum\_1= sum(TransactionValue))  
Transaction <- filter(data,sum\_1 >130000)  
Transaction

## # A tibble: 6 × 2  
## Country sum\_1  
## <chr> <dbl>  
## 1 Australia 137077.  
## 2 EIRE 263277.  
## 3 France 197404.  
## 4 Germany 221698.  
## 5 Netherlands 284662.  
## 6 United Kingdom 8187806.

4.The variable is read as a categorical when you read data 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(ORetail$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"

Now, let’s separate date, day of the week and hour components dataframe with names as New\_Invoice\_Date, Invoice\_Day\_Week and New\_Invoice\_Hour:

#Converting InvoiceDate to datetime format  
ORetail$InvoiceDate <- as.POSIXct(ORetail$InvoiceDate, format = "%Y-%m-%d %H:%M:%S")  
  
# Creating new columns for date, day of week, and hour  
ORetail$New\_Invoice\_Date <- as.Date(ORetail$InvoiceDate)  
ORetail$Invoice\_Day\_Week <- weekdays(ORetail$InvoiceDate)  
ORetail$New\_Invoice\_Hour <- format(ORetail$InvoiceDate, format = "%H:%M:%S")  
  
# View the first few rows of the updated dataset  
head(ORetail)

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

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.

#Create two example date values  
date1 <- as.Date("2023-08-15")  
date2 <- as.Date("2023-09-15")  
  
# Determine the number of days between the two dates  
Days\_between <- as.numeric(date2 - date1)  
Days\_between

## [1] 31

we can convert dates to days of the week also.So for that,let’s create a new variable.

ORetail$Invoice\_Day\_Week= weekdays(ORetail$New\_Invoice\_Date)

For the Hour, let’s just take the hour (ignore the minute) and convert into a normal numerical value.

ORetail$New\_Invoice\_Hour = as.numeric(format(Temp, "%H"))

Lets define the month as a separate numeric variable too:

ORetail$New\_Invoice\_Month = as.numeric(format(Temp, "%m"))

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

# calculate the total number of transactions for each day of the week  
day\_counts <- table(ORetail$Invoice\_Day\_Week)  
  
# calculate the percentage of transactions for each day of the week  
day\_percents <- round(100 \* prop.table(day\_counts), digits = 2)  
  
# combine the day counts and percents into a data frame  
day\_summary <- data.frame(Day = names(day\_counts),  
 TotalNumber = day\_counts,  
 Percentage = day\_percents)  
  
# display the resulting data frame  
day\_summary

## [1] Freq Percentage  
## <0 rows> (or 0-length row.names)

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

d1<-summarise(group\_by(ORetail,Invoice\_Day\_Week),Transaction\_Volume=sum(TransactionValue))  
d2<-mutate(d1,percentage=(Transaction\_Volume/sum(Transaction\_Volume))\*100)  
d2

## # A tibble: 1 × 3  
## Invoice\_Day\_Week Transaction\_Volume percentage  
## <chr> <dbl> <dbl>  
## 1 <NA> 9747748. 100

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

m1<-summarise(group\_by(ORetail,New\_Invoice\_Month),Transaction\_Volume=sum(TransactionValue))  
m2<-mutate(m1,percentage=(Transaction\_Volume/sum(Transaction\_Volume))\*100)  
m2

## # A tibble: 12 × 3  
## New\_Invoice\_Month Transaction\_Volume percentage  
## <dbl> <dbl> <dbl>  
## 1 1 560000. 5.74  
## 2 2 498063. 5.11  
## 3 3 683267. 7.01  
## 4 4 493207. 5.06  
## 5 5 723334. 7.42  
## 6 6 691123. 7.09  
## 7 7 681300. 6.99  
## 8 8 682681. 7.00  
## 9 9 1019688. 10.5   
## 10 10 1070705. 11.0   
## 11 11 1461756. 15.0   
## 12 12 1182625. 12.1

4.d) What was the date with the highest number of transactions from Australia?

ORetail <- ORetail %>% mutate(TransactionValue = Quantity \* UnitPrice)  
ORetail %>%filter(Country == 'Australia') %>%group\_by(New\_Invoice\_Date) %>%  
summarise(total\_transactions = sum(TransactionValue)) %>%  
arrange(desc(total\_transactions)) %>% slice(1)

## # A tibble: 1 × 2  
## New\_Invoice\_Date total\_transactions  
## <date> <dbl>  
## 1 NA 137077.

4.e) 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.

library(zoo)

##   
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

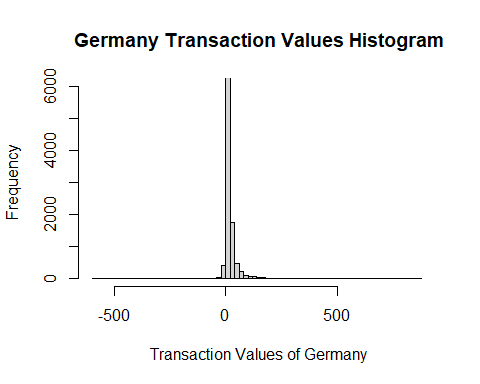
e1<-summarise(group\_by(ORetail,New\_Invoice\_Hour),Transaction\_min=n\_distinct(InvoiceNo))  
e1<-filter(e1,New\_Invoice\_Hour>=7&New\_Invoice\_Hour<=20)  
e12<-rollmax(e1$Transaction\_min,3,sum)  
e123<-which.min(e12)  
e123

## [1] 12

# Starting the work at 12 noon is suitable for maintenance.

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

Germany\_data <- subset(ORetail,Country == "Germany")  
hist(Germany\_data$TransactionValue,xlim = c(-600,900),breaks=100,xlab = "Transaction Values of Germany",ylim = c(0,6000),ylab = "Frequency",main = "Germany Transaction Values Histogram")

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

ORetail1 <- na.omit(ORetail)  
result1 <- summarise(group\_by(ORetail1,CustomerID), sum2= sum(TransactionValue))  
result1[which.max(result1$sum2),]

## # A tibble: 0 × 2  
## # ℹ 2 variables: CustomerID <int>, sum2 <dbl>

data2 <- table(ORetail$CustomerID)  
data2 <- as.data.frame(data2)  
result2 <- data2[which.max(data2$Freq),]  
result2

## Var1 Freq  
## 4043 17841 7983

1. Calculate the percentage of missing values for each variable in the dataset.

missing\_values <- colMeans(is.na(ORetail) \* 100)  
missing\_values

## InvoiceNo StockCode Description Quantity   
## 0.00000 0.00000 0.00000 0.00000   
## InvoiceDate UnitPrice CustomerID Country   
## 100.00000 0.00000 24.92669 0.00000   
## TransactionValue New\_Invoice\_Date Invoice\_Day\_Week New\_Invoice\_Hour   
## 0.00000 100.00000 100.00000 0.00000   
## New\_Invoice\_Month   
## 0.00000

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

ORetail2 <- ORetail %>% filter(is.na(CustomerID)) %>% group\_by(Country)  
summary(ORetail2$Country)

## Length Class Mode   
## 135080 character character

9.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).

library(dplyr)  
  
#convert InvoiceDate to proper format  
ORetail$InvoiceDate <- as.POSIXct(ORetail$InvoiceDate, format="%Y-%m-%d %H:%M:%S")  
  
# subset the data to include only CustomerID and InvoiceDate  
custo\_dates <- select(ORetail, CustomerID, InvoiceDate)  
  
# sort the data by CustomerID and InvoiceDate  
custo\_dates <- arrange(custo\_dates, CustomerID, InvoiceDate)  
  
# calculate time difference between consecutive shopping trips for each customer  
custo\_times <- group\_by(custo\_dates, CustomerID) %>%  
mutate(diff\_days = as.numeric(difftime(InvoiceDate, lag(InvoiceDate), units = "days")))  
  
# calculate the average time difference across all customers  
avg\_days\_between\_shopping <- mean(na.omit(custo\_times$diff\_days))  
avg\_days\_between\_shopping

## [1] NaN

1. In 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.With this definition, what is the return rate for the French customers?Consider the cancelled transactions as those where the ‘Quantity’ variable has a negative value.

ORetail\_table <- filter(ORetail, Country == "France")  
totalrow <- nrow(ORetail\_table)  
total\_transactions <- nrow(ORetail\_table)  
cancelled\_transactions <- nrow(filter(ORetail\_table, TransactionValue < 0))  
return\_rate <- cancelled\_transactions / total\_transactions  
return\_rate

## [1] 0.01741264

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 <- tapply(ORetail$TransactionValue, ORetail$StockCode, sum)  
product\_with\_highest\_revenue <- names(product\_revenue)[which.max(product\_revenue)]  
product\_with\_highest\_revenue

## [1] "DOT"

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

uniq\_custo <- unique(ORetail$CustomerID)  
number\_of\_uniq\_custo <- length(uniq\_custo)  
number\_of\_uniq\_custo

## [1] 4373