Assignment 2

Online\_Retail <- read.csv("C:/Users/abinaya/Downloads/Online\_Retail.csv")  
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

library(zoo)

##   
## Attaching package: 'zoo'

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

library(readxl)

1)Show the breakdown of the number of transactions by countries i.e., how many transactions are in the dataset for each country.Show this in total number and also in percentage. Show only countries accounting for more than 1% of the total transactions.

set.seed(123)  
Online\_Retail %>% group\_by(Country)%>% summarise(transactions = n())%>% mutate(percentage= (transactions/541909)\*100)%>% arrange(desc(transactions))%>% filter(data <- percentage > 1)

## # A tibble: 4 × 3  
## Country transactions percentage  
## <chr> <int> <dbl>  
## 1 United Kingdom 495478 91.4   
## 2 Germany 9495 1.75  
## 3 France 8557 1.58  
## 4 EIRE 8196 1.51

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

Online\_Retail<- mutate(Online\_Retail, "TransactionValue"=TransactionValue<- Online\_Retail$Quantity \* Online\_Retail$UnitPrice)  
colnames(Online\_Retail)

## [1] "InvoiceNo" "StockCode" "Description" "Quantity"   
## [5] "InvoiceDate" "UnitPrice" "CustomerID" "Country"   
## [9] "TransactionValue"

head(Online\_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

3)Using the newly created variable, TransactionValue,show the breakdown of transaction valuesby countries. Show this in total sum of transaction values. Show only countries with total transaction exceeding 130,000 British Pound.

Online\_Retail%>% group\_by(Country)%>% summarise(total.sum.of.transaction.values = sum(TransactionValue))%>% arrange(desc(total.sum.of.transaction.values))%>% filter(total.sum.of.transaction.values>130000)

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

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 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(Online\_Retail$InvoiceDate,format=‘%m/%d/%Y %H:%M’,tz=‘GMT’)

Check the variable using, head(Temp). 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:

Online\_Retail$New\_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:

Online\_Retail$New\_Invoice\_Month = as.numeric(format(Temp, “%m”))

#let’s convert ‘InvoiceDate’ into a POSIXltobject:  
Temp=strptime(Online\_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')  
#Now, let’s separate date, day of the week and hour components dataframe with names as   
#New\_Invoice\_Date,Invoice\_Day\_Weekand New\_Invoice\_Hour:  
Online\_Retail$New\_Invoice\_Date<-as.Date(Temp)  
#knowing two date values,the object allows you to know the difference between the two dates in terms of the number days.   
Online\_Retail$New\_Invoice\_Date[20000]-Online\_Retail$New\_Invoice\_Date[10]

## Time difference of 8 days

#Also we can convert dates to days of the week. Let’s define a new variable for that  
Online\_Retail$Invoice\_Day\_Week=weekdays(Online\_Retail$New\_Invoice\_Date)  
#For the Hour, let’s just take the hour (ignore the minute) and convert into a normal numerical value:  
Online\_Retail$New\_Invoice\_Hour =as.numeric(format(Temp,"%H"))  
#Finally, lets define the month as a separate numeric variable too:  
Online\_Retail$New\_Invoice\_Month = as.numeric(format(Temp, "%m"))

Now answer the flowing questions.

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

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

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

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

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.

# 4.a)  
Online\_Retail%>% group\_by(Invoice\_Day\_Week)%>% summarise(Number.of.transaction=(n()))%>% mutate(Number.of.transaction,'percent'=(Number.of.transaction\*100)/sum(Number.of.transaction))

## # A tibble: 6 × 3  
## Invoice\_Day\_Week Number.of.transaction percent  
## <chr> <int> <dbl>  
## 1 Friday 82193 15.2  
## 2 Monday 95111 17.6  
## 3 Sunday 64375 11.9  
## 4 Thursday 103857 19.2  
## 5 Tuesday 101808 18.8  
## 6 Wednesday 94565 17.5

# 4.b)  
Online\_Retail%>% group\_by(Invoice\_Day\_Week)%>% summarise(Volume.of.transaction=(sum(TransactionValue)))%>% mutate(Volume.of.transaction,'percent'=(Volume.of.transaction\*100)/sum(Volume.of.transaction))

## # A tibble: 6 × 3  
## Invoice\_Day\_Week Volume.of.transaction percent  
## <chr> <dbl> <dbl>  
## 1 Friday 1540611. 15.8   
## 2 Monday 1588609. 16.3   
## 3 Sunday 805679. 8.27  
## 4 Thursday 2112519 21.7   
## 5 Tuesday 1966183. 20.2   
## 6 Wednesday 1734147. 17.8

# 4.c)  
Online\_Retail%>% group\_by(New\_Invoice\_Month)%>%  
summarise(Volume.By.Month=sum(TransactionValue))%>% mutate(Volume.By.Month,'Percent'=(Volume.By.Month\*100)/sum(Volume.By.Month))

## # A tibble: 12 × 3  
## New\_Invoice\_Month Volume.By.Month Percent  
## <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  
b<-Online\_Retail%>% group\_by(New\_Invoice\_Date,Country)%>%  
filter(Country=='Australia')%>% summarise(Number=sum(Quantity),amount=sum(TransactionValue))%>% arrange(desc(Number))

## `summarise()` has grouped output by 'New\_Invoice\_Date'. You can override using  
## the `.groups` argument.

b

## # A tibble: 49 × 4  
## # Groups: New\_Invoice\_Date [49]  
## New\_Invoice\_Date Country Number amount  
## <date> <chr> <int> <dbl>  
## 1 2011-06-15 Australia 15241 23427.  
## 2 2011-08-18 Australia 12196 21880.  
## 3 2011-03-03 Australia 10162 16558.  
## 4 2011-02-15 Australia 8384 14023.  
## 5 2011-05-17 Australia 8268 11925.  
## 6 2011-10-05 Australia 7135 16472.  
## 7 2011-01-06 Australia 4802 7154.  
## 8 2011-07-13 Australia 4332 2796.  
## 9 2011-11-15 Australia 3130 5355.  
## 10 2011-09-01 Australia 2836 2942.  
## # … with 39 more rows

b<-b[b['Number']==max(b['Number']),]   
b

## # A tibble: 1 × 4  
## # Groups: New\_Invoice\_Date [1]  
## New\_Invoice\_Date Country Number amount  
## <date> <chr> <int> <dbl>  
## 1 2011-06-15 Australia 15241 23427.

# 4.e)  
f=Online\_Retail%>% group\_by(New\_Invoice\_Hour)%>% summarise(Total.transaction= n())  
n<-rollapply(f['Total.transaction'],2,sum)%>% index(min(n))  
n

## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14

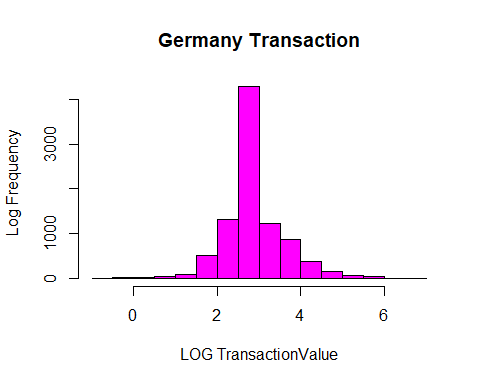
print('According to the data, the ideal time to shut down a website for two hours straight for maintenance is in the morning, between 7 and 9.')

## [1] "According to the data, the ideal time to shut down a website for two hours straight for maintenance is in the morning, between 7 and 9."

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

hist(x=log(Online\_Retail$TransactionValue[Online\_Retail$Country=="Germany"]),xlab = "LOG TransactionValue",col = 'magenta' ,main = 'Germany Transaction',ylab = 'Log Frequency')

## Warning in log(Online\_Retail$TransactionValue[Online\_Retail$Country ==  
## "Germany"]): NaNs produced



6)Which customer had the highest number of transactions? Which customer is most valuable

data\_1<- Online\_Retail %>% group\_by(CustomerID)%>%  
summarise(CustomerTransaction = n())%>% filter(CustomerID != "NA")%>% filter(CustomerTransaction ==max(CustomerTransaction) )  
print(paste('The customerID had the highest number of transactions is',data\_1$CustomerID,'with max transaction of ',data\_1$CustomerTransaction))

## [1] "The customerID had the highest number of transactions is 17841 with max transaction of 7983"

data\_2<- Online\_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) )  
print(paste('Most valuable customerID is',data\_2$CustomerID,'with total transaction Amount $',data\_2$total.transaction.by.each.customer))

## [1] "Most valuable customerID is 14646 with total transaction Amount $ 279489.02"

7)Calculate the percentage of missing values for each variable in the dataset. Hint colMeans():

Null\_Value<-colMeans(is.na(Online\_Retail))  
print(paste('Online customerID column has missing values in dataset and i.e.',Null\_Value['CustomerID']\*100,'% of whole data'))

## [1] "Online customerID column has missing values in dataset and i.e. 24.9266943342886 % of whole data"

8)What are the number of transactions with missing CustomerID records by countries

Online\_Retail%>% group\_by(Country)%>% filter(is.na(CustomerID))%>% summarise(No.of.missing.CustomerID=n())

## # A tibble: 9 × 2  
## Country No.of.missing.CustomerID  
## <chr> <int>  
## 1 Bahrain 2  
## 2 EIRE 711  
## 3 France 66  
## 4 Hong Kong 288  
## 5 Israel 47  
## 6 Portugal 39  
## 7 Switzerland 125  
## 8 United Kingdom 133600  
## 9 Unspecified 202

9)On average, how often the costumers comeback to the website for their next shopping Hint: 1. A close approximation is also acceptable and you may find diff() function useful.

Averg<-Online\_Retail%>% group\_by(CustomerID)%>%  
summarise(difference.in.consecutivedays= diff(New\_Invoice\_Date))%>%  
filter(difference.in.consecutivedays>0)

## `summarise()` has grouped output by 'CustomerID'. You can override using the  
## `.groups` argument.

print(paste('The average number of days between consecutive shopping is',mean(Averg$difference.in.consecutivedays)))

## [1] "The average number of days between consecutive shopping is 38.4875"

10)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 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.

Return\_value<-nrow(Online\_Retail%>% group\_by(CustomerID)%>% filter((Country=='France')&(TransactionValue<0)&(CustomerID != 'Na')))  
total\_french\_customer<-nrow(Online\_Retail%>%  
group\_by(CustomerID)%>% filter((Country=='France')&(CustomerID != 'Na')))  
   
print(paste('Return rate for french customer is given as',((Return\_value)/(total\_french\_customer))\*100,'percent'))

## [1] "Return rate for french customer is given as 1.75479919915204 percent"

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

Total\_customer1<-Online\_Retail%>%  
group\_by(Description,StockCode)%>%  
summarise(n=sum(TransactionValue))%>%  
arrange(desc(n))

## `summarise()` has grouped output by 'Description'. You can override using the  
## `.groups` argument.

x<- Total\_customer1[Total\_customer1['n']==max(Total\_customer1['n']),]  
x

## # A tibble: 1 × 3  
## # Groups: Description [1]  
## Description StockCode n  
## <chr> <chr> <dbl>  
## 1 DOTCOM POSTAGE DOT 206245.

print(paste('The highest revenue generated product is', x$Description,'with stock code',x$StockCode))

## [1] "The highest revenue generated product is DOTCOM POSTAGE with stock code DOT"

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

print(paste('Total no. of customers with valid customer id are ',length(unique(Online\_Retail$CustomerID))-1,'. This does not include null CustomerID'))

## [1] "Total no. of customers with valid customer id are 4372 . This does not include null CustomerID"