Assignment_1

SIVAREDDY

2022-03-13

```
library(readr)
Online_Retail <- read_csv("Online_Retail.csv")
```

```
## Rows: 541909 Columns: 8
## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr (5): InvoiceNo, StockCode, Description, InvoiceDate, Country
## dbl (3): Quantity, UnitPrice, CustomerID
##
## 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 m
essage.
```

```
View(Online_Retail)
```

#Question -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.

Ans.

```
country_totaltransaction <- table(Online_Retail$Country)
transaction_percent<- round(100*prop.table(country_totaltransaction))
percentage <- cbind(country_totaltransaction, transaction_percent)
Question1_solution <-subset(percentage, transaction_percent >1)
Question1_solution
```

```
##                country_totaltransaction transaction_percent
## EIRE                               8196                   2
## France                             8557                   2
## Germany                            9495                   2
## United Kingdom                   495478                  91
```

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

Ans

Creating new variable

```
Transactionvalue <- c(Online_Retail$Quantity * Online_Retail$UnitPrice)
Online_Retail$Transactionvalue = Transactionvalue
head(Online_Retail)

## # A tibble: 6 x 9
##   InvoiceNo StockCode Description      Quantity InvoiceDate UnitPrice Cust
omerID
##   <chr>     <chr>     <chr>               <dbl> <chr>           <dbl>
<dbl>
## 1 536365    85123A    WHITE HANGING H~        6 12/1/2010 ~      2.55
17850
## 2 536365    71053     WHITE METAL LAN~        6 12/1/2010 ~      3.39
17850
## 3 536365    84406B    CREAM CUPID HEA~        8 12/1/2010 ~      2.75
17850
## 4 536365    84029G    KNITTED UNION F~        6 12/1/2010 ~      3.39
17850
## 5 536365    84029E    RED WOOLLY HOTT~        6 12/1/2010 ~      3.39
17850
## 6 536365    22752     SET 7 BABUSHKA ~        2 12/1/2010 ~      7.65
17850
## # ... with 2 more variables: Country <chr>, Transactionvalue <dbl>
```

#Question-3 #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.

Ans

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.1.3

##
## 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
```

```
Total_Transactionvalue <- sum(Transactionvalue)
data <-summarise(group_by(Online_Retail, Country), Total_Transactionvalue)
Transactionvalue_1 <- filter(data,  Total_Transactionvalue>130000)
Transactionvalue_1
```

```
## # A tibble: 38 x 2
##    Country         Total_Transactionvalue
##    <chr>                            <dbl>
##  1 Australia                     9747748.
##  2 Austria                       9747748.
##  3 Bahrain                       9747748.
##  4 Belgium                       9747748.
##  5 Brazil                        9747748.
##  6 Canada                        9747748.
##  7 Channel Islands               9747748.
##  8 Cyprus                        9747748.
##  9 Czech Republic                9747748.
## 10 Denmark                       9747748.
## # ... with 28 more rows
```

# Question-4 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.

Ans

```
Temp=strptime(Online_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"
```

```
Online_Retail$New_Invoice_Date <- as.Date(Temp)


Online_Retail$New_Invoice_Date[20000]- Online_Retail$New_Invoice_Date[10]
```

```
## Time difference of 8 days
```

```
Invoice_Day_Week =  weekdays(Online_Retail$New_Invoice_Date)
Online_Retail$Invoice_Day_Week= Invoice_Day_Week



Online_Retail$New_Invoice_Hour = as.numeric(format(Temp, "%H"))
New_Invoice_Hour = Online_Retail$New_Invoice_Hour
```

```
Online_Retail$New_Invoice_Month = as.numeric(format(Temp, "%m"))
New_Invoice_Month = Online_Retail$New_Invoice_Month

Online_Retail$New_Invoice_Year = as.numeric(format(Temp, "%y"))
New_Invoice_Year<- Online_Retail$New_Invoice_Year
```

```
#4a Show the percentage of transactions (by numbers) by days of the week (ext
ra 2 marks)
```

```
Online_Retail %>% select(Invoice_Day_Week,Quantity) %>% filter( Invoice_Day_W
eek %in%
c("Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday")) %>
% count(Invoice_Day_Week)
```

```
## # A tibble: 6 x 2
##   Invoice_Day_Week      n
##   <chr>             <int>
## 1 Friday            82193
## 2 Monday            95111
## 3 Sunday            64375
## 4 Thursday         103857
## 5 Tuesday          101808
## 6 Wednesday         94565
```

```
All_Transaction <- length(Online_Retail$Quantity)
All_Transaction
```

```
## [1] 541909
```

```
# Sunday percent

Sunday=64375

Sunday_Percentage <- (Sunday/All_Transaction)



## Monday percent

Monday=95111
Monday_Percentage <- Monday/All_Transaction

#Tuesday percent
Tuesday = 101808
Tuesday_Percentage<- Tuesday/All_Transaction

##Wednesday percent
Wednesday = 94565
Wednesday_Percentage <- Wednesday/All_Transaction
```

```
#Thursday percent
Thursday = 103857
Thursday_Percentage <- Thursday/All_Transaction

#Friday percent
Friday = 82193
Friday_Percentage <- Friday/All_Transaction

#Saturday Percent
Saturday = 0
Saturday_Percentage <- Saturday/All_Transaction

data.frame(Sunday_Percentage,Monday_Percentage,Tuesday_Percentage,Wednesday_P
ercentage,Thursday_Percentage,Friday_Percentage,Saturday_Percentage)

##   Sunday_Percentage Monday_Percentage Tuesday_Percentage Wednesday_Percent
age
## 1          0.118793          0.175511          0.1878692            0.1745
035
##   Thursday_Percentage Friday_Percentage Saturday_Percentage
## 1           0.1916503         0.1516731                   0
```

#4b Show the percentage of transactions (by transaction volume) by days of the week (extra 1 marks)

```
Transaction2<- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>%
filter(Invoice_Day_Week=="Sunday")
sum_sunday<- sum(Transaction2$Quantity)
sum_sunday

## [1] 467732

Transaction2<- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>%
filter(Invoice_Day_Week=="Monday")
sum_monday<- sum(Transaction2$Quantity)
sum_monday

## [1] 815354

Transaction2<-  Online_Retail %>% select(Invoice_Day_Week,Quantity) %>%
filter(Invoice_Day_Week=="Tuesday")
sum_tuesday<- sum(Transaction2$Quantity)
sum_tuesday

## [1] 961543

Transaction2<- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>%
filter(Invoice_Day_Week=="Wednesday")
sum_wednesday<- sum(Transaction2$Quantity)
sum_wednesday
```

```
## [1] 969558
```

```
Transaction2<- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>%
filter(Invoice_Day_Week=="Thursday")
sum_thursday<- sum(Transaction2$Quantity)
sum_thursday
```

```
## [1] 1167823
```

```
Transaction2<- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>%
filter(Invoice_Day_Week=="Friday")
sum_friday<- sum(Transaction2$Quantity)
sum_friday
```

```
## [1] 794440
```

```
Transaction2<- Online_Retail %>% select(Invoice_Day_Week,Quantity) %>%
filter(Invoice_Day_Week=="Saturday")
sum_saturday<- sum(Transaction2$Quantity)
sum_saturday
```

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

```
data.frame(sum_sunday,sum_monday,sum_tuesday,sum_wednesday,sum_thursday,sum_f
riday,sum_saturday)
```

```
##   sum_sunday sum_monday sum_tuesday sum_wednesday sum_thursday sum_friday
## 1     467732     815354      961543        969558      1167823     794440
##   sum_saturday
## 1            0
```

#4C Show the percentage of transactions (by transaction volume) by month of the year

```
Trans_volume<- sum(Online_Retail$Quantity)
Trans_volume
```

```
## [1] 5176450
```

```
percent_sunday<- sum_sunday/Trans_volume
percent_sunday
```

```
## [1] 0.09035768
```

```
percent_monday<- sum_monday/Trans_volume
percent_monday
```

```
## [1] 0.1575122
```

```
percent_tuesday<- sum_tuesday/Trans_volume
percent_tuesday
```

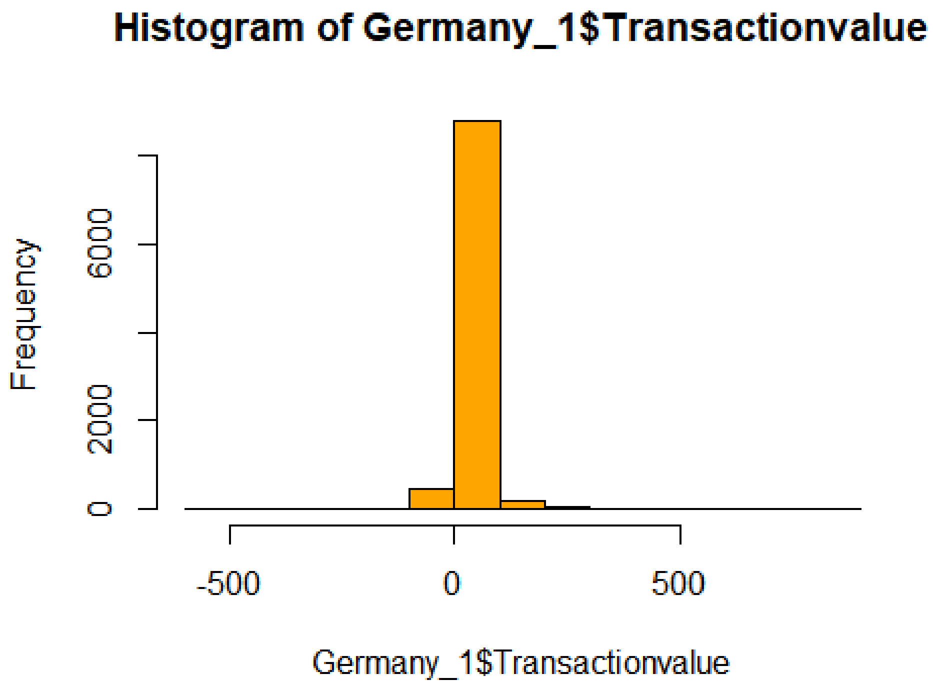
```
## [1] 0.1857534
```

```
percent_wednesday<- sum_wednesday/Trans_volume
percent_wednesday

## [1] 0.1873017

percent_thursday<- sum_thursday/Trans_volume
percent_thursday

## [1] 0.2256031

percent_friday<- sum_thursday/Trans_volume
percent_friday

## [1] 0.2256031

percentage_saturday<- sum_thursday/Trans_volume

data.frame(percent_sunday,percent_monday,percent_tuesday,percent_wednesday,pe
rcent_thursday,percent_friday,percentage_saturday)

##   percent_sunday percent_monday percent_tuesday percent_wednesday
## 1     0.09035768      0.1575122       0.1857534         0.1873017
##   percent_thursday percent_friday percentage_saturday
## 1        0.2256031      0.2256031           0.2256031

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

A <- Online_Retail %>% select(InvoiceDate,Quantity,Transactionvalue,Country)
%>% filter(Country == "Australia") %>% count(InvoiceDate)
A

## # A tibble: 66 x 2
##    InvoiceDate         n
##    <chr>           <int>
##  1 1/10/2011 9:58      1
##  2 1/11/2011 9:47     19
##  3 1/14/2011 11:36     3
##  4 1/17/2011 11:12    19
##  5 1/19/2011 9:13     13
##  6 1/20/2011 12:11     4
##  7 1/28/2011 14:37    20
##  8 1/6/2011 11:12     46
##  9 1/6/2011 12:37      2
## 10 10/5/2011 12:35     1
## # ... with 56 more rows

#4e 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 dist
ribution is at minimum for the customers? The responsible IT team is availabl
e from 7:00 to 20:00 every day.
```

```
library(zoo)

## Warning: package 'zoo' was built under R version 4.1.3

##
## Attaching package: 'zoo'

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

Question_e<-summarise(group_by(Online_Retail,New_Invoice_Hour),Transaction_min=n_distinct(InvoiceNo))
Question_e1<-filter(Question_e,New_Invoice_Hour>=7&New_Invoice_Hour<=20)
Question_e2<-rollapply(Question_e1$Transaction_min,3,sum)
Question_e3<-which.min(Question_e2)
Question_e3

## [1] 12
```

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

```
Germany_1<- c(Online_Retail %>% select(Transactionvalue,Country) %>% filter(Country=="Germany"))
hist(Germany_1$Transactionvalue, col ="Orange")
```



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

Ans.

customer having highest number of transactions and High valued Customer before removing NA values

```
customer_highnumber_transaction<- Online_Retail %>% select(CustomerID,Quantit
y,Transactionvalue) %>%count(CustomerID)
which.max(customer_highnumber_transaction$n)

## [1] 4373

customer_highnumber_transaction ["4373",]

## # A tibble: 1 x 2
##   CustomerID      n
##        <dbl>  <int>
## 1         NA 135080
```

High valued Customer before removing NA values

```
highvalued_customer <- group_by(Online_Retail, CustomerID) %>% summarize(tran
svalue_customer = sum(Transactionvalue))
which.max(highvalued_customer$transvalue_customer)

## [1] 4373

highvalued_customer["4373",]

## # A tibble: 1 x 2
##   CustomerID transvalue_customer
##        <dbl>               <dbl>
## 1         NA            1447682.
```

customer having highest number of transactions and High valued Customer after removing NA values

```
customer_highnumber_transaction<- na.omit(Online_Retail %>% select(CustomerID
,Quantity,Transactionvalue) %>% count(CustomerID))
which.max(customer_highnumber_transaction$n)

## [1] 4043

customer_highnumber_transaction ["4043",]

## # A tibble: 1 x 2
##   CustomerID     n
##        <dbl> <int>
## 1      17841  7983
```

High valued Customer after removing NA

```
highvalued_customer <- na.omit (group_by(Online_Retail, CustomerID) %>% summa
rize(transvalue_customer = sum(Transactionvalue)))
which.max(highvalued_customer$transvalue_customer)

## [1] 1704

highvalued_customer["1704",]

## # A tibble: 1 x 2
##   CustomerID transvalue_customer
##        <dbl>               <dbl>
## 1      14646             279489.
```

#7 Calculate the percentage of missing values for each variable in the dataset (5 marks).

```
colMeans(is.na(Online_Retail))

##          InvoiceNo          StockCode        Description           Quantity
##        0.000000000        0.000000000        0.002683107        0.000000000
##        InvoiceDate          UnitPrice         CustomerID            Country
##        0.000000000        0.000000000        0.249266943        0.000000000
##   Transactionvalue   New_Invoice_Date   Invoice_Day_Week   New_Invoice_Hour
##        0.000000000        0.000000000        0.000000000        0.000000000
## New_Invoice_Month   New_Invoice_Year
##        0.000000000        0.000000000
```

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

```
Online_Retail %>% select(Country,CustomerID) %>% filter(is.na(Online_Retail$C
ustomerID)) %>% count(Country)

## # A tibble: 9 x 2
##   Country             n
##   <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?

#10In 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? (10 marks). Consider the cancelled transactions as those where the 'Quantity' variable has a negative value.

```
Retail_table <- filter(Online_Retail,Country=="France")
totalrow <- nrow(Retail_table)

cancel <- nrow(subset(Retail_table,Transactionvalue<0))
cancel

## [1] 149

notcancel <- totalrow-cancel
notcancel

## [1] 8408

Total_value = (cancel + notcancel)

canceloftotal_retail=(cancel/Total_value)
canceloftotal_retail

## [1] 0.01741264
```

#11 What is the product that has generated the highest revenue for the retailer? (i.e. item with the

```
Product <- (group_by(Online_Retail, Description) %>% summarize( Product =
sum(Transactionvalue)))

which.max(Product$Product)

## [1] 1128

Product["1128",]

## # A tibble: 1 x 2
##   Description    Product
##   <chr>            <dbl>
## 1 DOTCOM POSTAGE 206245.
```

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

```
unique_customer<- sapply(Online_Retail, function(Online_Retail) length(unique
(Online_Retail)))
unique_customer
```

```
##         InvoiceNo         StockCode       Description          Quantity 
##             25900              4070              4212               722 
##       InvoiceDate         UnitPrice        CustomerID           Country 
##             23260              1630              4373                38 
##  Transactionvalue  New_Invoice_Date  Invoice_Day_Week  New_Invoice_Hour 
##              6204               305                 6                15 
## New_Invoice_Month  New_Invoice_Year 
##                12                 2
```

```
uniquecustomer_ID <- length(unique(Online_Retail$CustomerID))
uniquecustomer_ID
```

```
## [1] 4373
```