Assignment 1

Tejasvini Mavuleti

3/13/2022

```
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(readxl)
library(readr)
retail <- read.csv("C:/Users/mavul/Downloads/Online_Retail.csv")
colnames(retail)

## [1] "InvoiceNo"   "StockCode"   "Description" "Quantity"    "InvoiceDate"
## [6] "UnitPrice"   "CustomerID"  "Country"

nrow(retail)

## [1] 541909
```

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.

```
retail %>% group_by(Country) %>% summarise(transactions = n(), percentage =
(transactions/541909)*100 ) %>% filter(percentage>1)

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

2. Create a new variable 'TransactionValue' that is the product of the exising 'Quantity' and 'UnitPrice' variables.

```
retail["TransactionValue"] <- retail$Quantity* retail$UnitPrice
View(retail)
```

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.

```
retail %>% group_by(Country) %>% summarise(total=sum(TransactionValue))
```

```
## # A tibble: 38 x 2
##    Country           total
##    <chr>             <dbl>
##  1 Australia       137077.
##  2 Austria          10154.
##  3 Bahrain            548.
##  4 Belgium          40911.
##  5 Brazil            1144.
##  6 Canada            3666.
##  7 Channel Islands  20086.
##  8 Cyprus           12946.
##  9 Czech Republic     708.
## 10 Denmark          18768.
## # ... with 28 more rows
```

```
transactionexceeding <-retail %>% group_by(Country) %>%
summarise(total=sum(TransactionValue)) %>% filter(total>130000)
View(transactionexceeding)
```

converting InvoiceDate into a POSIXlt object:

```
Temp=strptime(retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')
retail$New_Invoice_Date <- as.Date(Temp)
retail$New_Invoice_Date[20000]- retail$New_Invoice_Date[10]
```

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

```
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"))
```

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

```
retail %>% group_by(Invoice_Day_Week) %>% summarise(count=n())%>%
mutate(percentage= (count/nrow(retail)*100))
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week  count percentage
##   <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
```

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

```
retail %>%
group_by(Invoice_Day_Week)%>%summarise(total=sum(TransactionValue))%>%mutate(
percentage=total/sum(total)*100)
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week    total percentage
##   <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
```

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

```
retail %>%
group_by(New_Invoice_Month)%>%summarise(total=sum(TransactionValue))%>%mutate
(percentage=total/sum(total)*100)
```

```
## # A tibble: 12 x 3
##    New_Invoice_Month    total 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
```

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

ANS: By observing the tibble, we can get that the maximum no.of transactions from Australia was 139 on 2011-06-15.

```
retail %>% group_by(New_Invoice_Date) %>% filter(Country == "Australia")  %>%
tally(sort= TRUE)
```

```
## # A tibble: 49 x 2
##    New_Invoice_Date     n
##    <date>           <int>
##  1 2011-06-15         139
##  2 2011-07-19         137
##  3 2011-08-18          97
##  4 2011-03-03          84
##  5 2011-10-05          82
##  6 2011-05-17          73
##  7 2011-02-15          69
##  8 2011-01-06          48
##  9 2011-07-14          35
## 10 2011-09-16          34
## # ... with 39 more rows
```

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?

```
retail %>%
  filter(New_Invoice_Hour>= 7 & New_Invoice_Hour<=20) %>%
group_by(New_Invoice_Hour) %>%
  tally(sort = TRUE) %>% arrange(n)
```

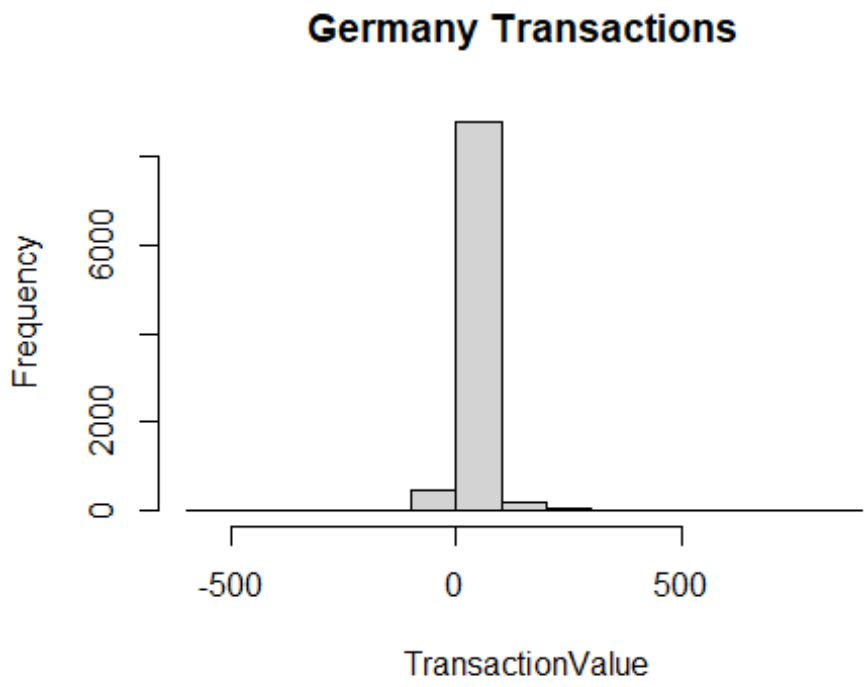
```
## # A tibble: 14 x 2
##    New_Invoice_Hour     n
##               <dbl> <int>
##  1                7   383
##  2               20   871
##  3               19  3705
##  4               18  7974
```

```
##  5                8  8909
##  6               17 28509
##  7                9 34332
##  8               10 49037
##  9               16 54516
## 10               11 57674
## 11               14 67471
## 12               13 72259
## 13               15 77519
## 14               12 78709
```

ANS: By observing the table, the 19th , 20th are the two consecutive hours which has the lowest sum of two consecutive hours.

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

```
retail%>% filter(Country == "Germany") %>% summary(total =
sum(TransactionValue))-> Germany
hist(x=(retail$TransactionValue[retail$Country=="Germany"]),xlab = "
TransactionValue",main = 'Germany Transactions',ylab = ' Frequency')
```



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

```
retail %>% group_by(CustomerID) %>% tally(sort = TRUE) %>%
filter(!is.na(CustomerID)) %>% filter(n==max(n))
```

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

```
retail%>% group_by(CustomerID) %>% summarise(highesttotalsumoftransactions =
sum(TransactionValue))%>% arrange(desc(highesttotalsumoftransactions))%>%
filter(CustomerID != "NA")%>%
  filter(highesttotalsumoftransactions ==max(highesttotalsumoftransactions) )
```

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

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

```
Missingvalues <- colMeans(is.na(retail)*100)
View(Missingvalues)
```

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

```
nrow(retail[is.na(retail$CustomerID),])
```

```
## [1] 135080
```

```
retail[is.na(retail$CustomerID),] %>% group_by(Country) %>%
summarise(missingcustomerID = n())
```

```
## # A tibble: 9 x 2
##   Country        missingcustomerID
##   <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?

```
retail%>% group_by(CustomerID)%>% summarise(avg_no_of_days=
diff(New_Invoice_Date)) %>% filter(avg_no_of_days>0)
```

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

```
## # A tibble: 15,200 x 2
## # Groups:   CustomerID [2,992]
##    CustomerID avg_no_of_days
##         <int> <drtn>
##  1      12347  50 days
##  2      12347  71 days
##  3      12347  63 days
##  4      12347  54 days
##  5      12347  90 days
##  6      12347  37 days
##  7      12348  40 days
##  8      12348  70 days
##  9      12348 173 days
## 10      12352  13 days
## # ... with 15,190 more rows
```

```
mean(retail$avg_no_of_days)
```

```
## Warning in mean.default(retail$avg_no_of_days): argument is not numeric or
## logical: returning NA
```

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

10. In the retail sector, it is very important to understand the return rate of the goods purchased by customers. what is the return rate for the French customers?

```
return_val<-nrow(retail%>% group_by(CustomerID)%>%
filter((Country=='France')&(TransactionValue<0)&(CustomerID != 'Na')))
total_french_customer<-nrow(retail%>% group_by(CustomerID)%>%
filter((Country=='France')&(CustomerID != 'Na')))
print(paste('Return rate for french customer
is',((return_val)/(total_french_customer))*100,'%'))
```

```
## [1] "Return rate for french customer is 1.75479919915204 %"
```

11. What is the product that has generated the highest revenue for the retailer?

```
retail %>% group_by(Description) %>% summarise(total=sum(TransactionValue))
%>% filter(total == max(total))
```

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

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

```
length(unique(retail$CustomerID))
```

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