BA Assignment_2

Abhinay Chiranjeeth Marneni

2022-10-26

Loading package

```
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

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

```
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
```

Import the Online_Retail.csv dataset

```
Online_Retail<-read.csv("C:/Users/abhin/OneDrive/Documents/Assigments Buss 1sem/BA/Ass2/Online_Retail.c
```

1.

```
summary(Online_Retail) # Summarizing onlineretail
```

```
##    InvoiceNo          StockCode          Description          Quantity
##   Length:541909      Length:541909      Length:541909       Min.   :-80995.00
##   Class :character   Class :character   Class :character    1st Qu.:      1.00
##   Mode  :character   Mode  :character   Mode  :character    Median :      3.00
##                                                             Mean   :      9.55
##                                                             3rd Qu.:     10.00
##                                                             Max.   : 80995.00
##
##    InvoiceDate          UnitPrice             CustomerID          Country
##   Length:541909      Min.   :-11062.06     Min.   :12346      Length:541909
##   Class :character   1st Qu.:      1.25     1st Qu.:13953      Class :character
##   Mode  :character   Median :      2.08     Median :15152      Mode  :character
##                      Mean   :      4.61     Mean   :15288
##                      3rd Qu.:      4.13     3rd Qu.:16791
##                      Max.   : 38970.00     Max.   :18287
##                                            NA's   :135080
```

```r
summary(Online_Retail$Country) # Summarizing onlineretailcountry
```

```
##    Length      Class       Mode
##    541909  character  character
```

```r
country_totaltransaction<-table(Online_Retail$Country)
transaction_percent<-round(100*prop.table(country_totaltransaction),digits = 2)
percentage<-cbind(country_totaltransaction,transaction_percent)
total<-subset(percentage,transaction_percent>1.0)# the represent all countries less than 1% value of to
total
```

```
##                  country_totaltransaction transaction_percent
## EIRE                                 8196                1.51
## France                               8557                1.58
## Germany                              9495                1.75
## United Kingdom                     495478               91.43
```

2.

```r
Online_Retail <- Online_Retail %>% mutate(TransactionValue= Quantity * UnitPrice)
summary(Online_Retail$TransactionValue)
```

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

3.

```r
data <- summarise(group_by(Online_Retail,Country),sum_value= sum(TransactionValue)) # group the transac
Transaction <- filter(data,sum_value >130000) # filter out countries which spend less than 130000
Transaction
```

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

4.

```
Temp=strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT') # read  temporary data varaia
head(Temp)  # Checking  the  variable by  using,  head(Temp) command
```

```
## [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)  # create new invoice for date
```

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

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

```
Online_Retail$Invoice_Day_Week= weekdays(Online_Retail$New_Invoice_Date) # converting new invoice date
```

```
Online_Retail$New_Invoice_Hour = as.numeric(format(Temp, "%H")) # create new invoice hours
```

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

a)

```
a<-summarise(group_by(Online_Retail,Invoice_Day_Week),Transaction_Value=n_distinct(InvoiceNo))
a1<-mutate(a, transaction_percent=(Transaction_Value/sum(Transaction_Value))*100)
a1
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Transaction_Value transaction_percent
##   <chr>                        <int>               <dbl>
## 1 Friday                        4184                16.2
## 2 Monday                        4138                16.0
## 3 Sunday                        2381                 9.19
## 4 Thursday                      5660                21.9
## 5 Tuesday                       4722                18.2
## 6 Wednesday                     4815                18.6
```

b)

```
b<-summarise(group_by(Online_Retail,Invoice_Day_Week),Transaction_Volume=sum(TransactionValue))
b1<-mutate(b,percentage=(Transaction_Volume/sum(Transaction_Volume))*100)
b1
```

```
## # A tibble: 6 x 3
##   Invoice_Day_Week Transaction_Volume 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)

```
c<-summarise(group_by(Online_Retail,New_Invoice_Month),Transaction_Volume=sum(TransactionValue))
c1<-mutate(c,percentage=(Transaction_Volume/sum(Transaction_Volume))*100)
c1
```

```
## # A tibble: 12 x 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
```

d)

```
Online_Retail <- Online_Retail %>% mutate(TransactionValue= Quantity * UnitPrice)
Online_Retail %>% filter(Country == 'Australia') %>% group_by(New_Invoice_Date) %>% summarise(max=max(T
```

```
## # A tibble: 49 x 2
##    New_Invoice_Date    max
##    <date>            <dbl>
```

```
##  1 2010-12-01             51
##  2 2010-12-08             71.4
##  3 2010-12-14             -6.25
##  4 2010-12-17            148.
##  5 2011-01-06           1020
##  6 2011-01-10             81.6
##  7 2011-01-11             35.4
##  8 2011-01-14            142.
##  9 2011-01-17             47.4
## 10 2011-01-19             38.2
## # ... with 39 more rows
```

e)

```
e<-summarise(group_by(Online_Retail,New_Invoice_Hour),Transaction_min=n_distinct(InvoiceNo))
e1<-filter(e,New_Invoice_Hour>=7&New_Invoice_Hour<=20)
e2<-rollapply(e1$Transaction_min,3,sum)
e3<-which.min(e2)
e3
```

```
## [1] 12
```

5.

```
Germany_data <- subset(Online_Retail$TransactionValue,Online_Retail$Country == "Germany")
hist(Germany_data, xlim = c (-600, 900), breaks = 100 , xlab = "Transaction Values of Germany", main =
```

Germany

6.
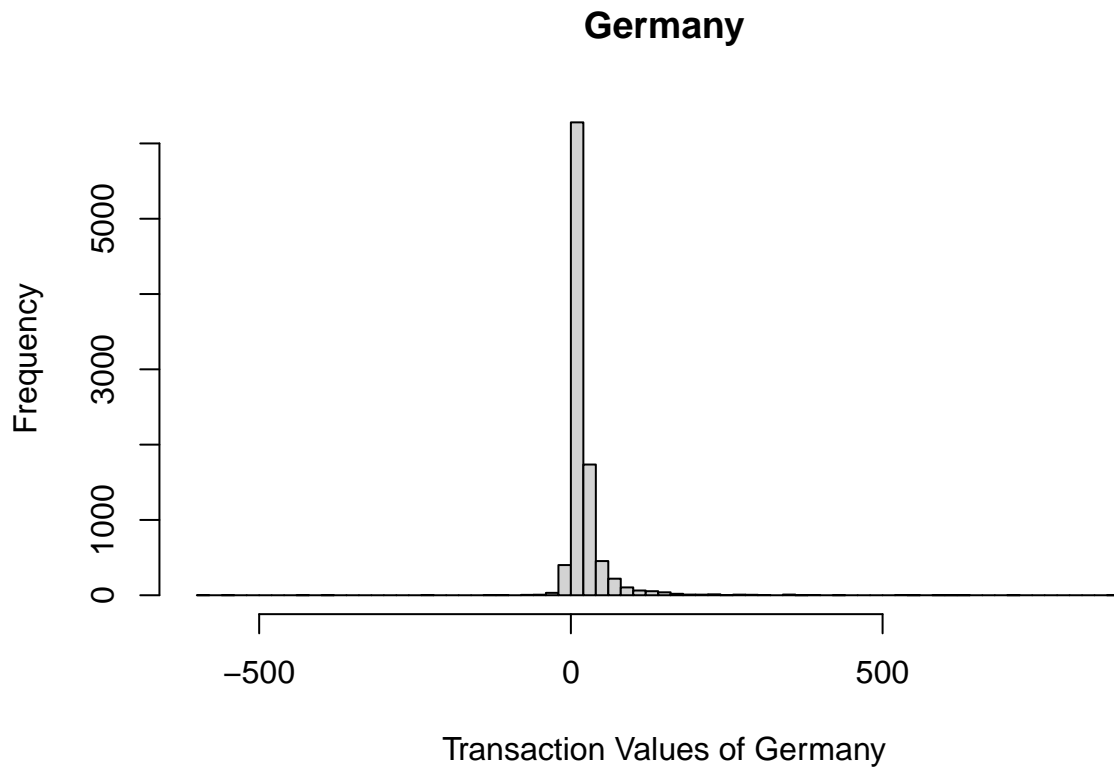
```r
Online_Retail1 <- na.omit(Online_Retail)
result <- summarise(group_by(Online_Retail1,CustomerID), sum2= sum(TransactionValue))
result[which.max(result$sum2),]
```

```
## # A tibble: 1 x 2
##   CustomerID    sum2
##        <int>   <dbl>
## 1      14646 279489.
```

```r
data2 <- table(Online_Retail$CustomerID)
data2 <- as.data.frame(data2)
result1 <- data2[which.max(data2$Freq),] # group and summarize the highest transactions of coustmer
result1
```

```
##        Var1 Freq
## 4043 17841 7983
```

7.

```r
missing_values <- colMeans(is.na(Online_Retail)*100)
missing_values # Calculate the missing values for each variable in dataset
```

```
##         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
```

8.

```r
Online_Retail2 <- Online_Retail %>% filter(is.na(CustomerID)) %>% group_by(Country)
summary(Online_Retail2$Country) # filter and summarize the NA and total values of dataset
```

```
##     Length      Class       Mode
##     135080  character  character
```

9.

```r
Online_Retail_NA_Removed <- na.omit(Online_Retail)
Online_Retail_NA_Neg_Removed <- subset(Online_Retail_NA_Removed, Quantity > 0)
Online_Retail_Subset <- Online_Retail_NA_Neg_Removed[,c("CustomerID","New_Invoice_Date")]
Online_Retail_Subset_Distinct <- distinct(Online_Retail_Subset)
Online_Retail_Subset_Distinct %>%
  group_by(CustomerID) %>%
  arrange(New_Invoice_Date) %>%
  summarise(avg = mean(diff(New_Invoice_Date))) %>%
  na.omit() %>%
  summarise(avg_days_between_shopping = mean(avg))
```

```
## # A tibble: 1 x 1
##   avg_days_between_shopping
##   <drtn>
## 1 78.42025 days
```

10.

```r
Online_Retail_table <- filter(Online_Retail,Country=="France")
totalrow <- nrow(Online_Retail_table)
cancel <- nrow(subset(Online_Retail_table,TransactionValue<0))
cancel
```

```
## [1] 149
```

```r
notcancel <- totalrow-cancel
notcancel
```

```
## [1] 8408
```

```r
TEST2=(cancel/8556)
TEST2 # calculate the return rate and total transaction for France
```

```
## [1] 0.01741468
```

11.

```r
Transaction_Value <- tapply(Online_Retail$TransactionValue, Online_Retail$StockCode  , sum)
Transaction_Value[which.max(Transaction_Value)] # generate the highest revenue for retailer products
```

```
##      DOT
## 206245.5
```

12.

```r
unique_customers <- unique(Online_Retail$CustomerID)
length(unique_customers) # removing the duplicate entries of customerID
```

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