

---

itle: "Assignment 2"  
uthor: "Snehitha Anpur"  
ate: "2022-10-28"  
utput: pdf\_document

---

```
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
```

```
OnlineRetail=read.csv("D:\\MSBA\\rTutorial\\Rtutorial\\Online_Retail.csv")
```

Question 1:

Showing the Count of transactions by country and the percentage of transactions by country greater than 1

```
Countries_count = OnlineRetail %>% group_by(Country) %>% count(Country)  
  
Countries_pct = OnlineRetail %>% group_by(Country) %>% summarise(percent = 100* n()/nrow(OnlineRetail))  
  
Fltrd_Cntry_pct = filter(Countries_pct, percent>1)  
  
#Countries Count  
Countries_count
```

```
## # A tibble: 38 x 2  
## # Groups:   Country [38]  
##   Country      n  
##   <chr>      <int>  
## 1 Australia    1259  
## 2 Austria       401  
## 3 Bahrain       19  
## 4 Belgium     2069  
## 5 Brazil        32  
## 6 Canada       151  
## 7 Channel Islands 758  
## 8 Cyprus       622  
## 9 Czech Republic  30  
## 10 Denmark     389  
## # ... with 28 more rows
```

```
#Percentage of transactions greater than 1
Fltrd_Cntry_pct
```

```
## # A tibble: 4 x 2
##   Country      percent
##   <chr>         <dbl>
## 1 EIRE          1.51
## 2 France        1.58
## 3 Germany       1.75
## 4 United Kingdom 91.4
```

Question 2:

```
TransactionValue = (OnlineRetail$Quantity * OnlineRetail$UnitPrice)

#Adding the TransactionValue column to the OnlineRetail table
Online_Retail = cbind(OnlineRetail,TransactionValue)
```

Question 3:

```
Trans_sum = Online_Retail %>% group_by(Country) %>% summarise(sum=sum(TransactionValue))

Fltrd_Trans_sum = filter(Trans_sum,Trans_sum$sum>130000)

#Sum of TransactionValue for each countries
Trans_sum
```

```
## # A tibble: 38 x 2
##   Country      sum
##   <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
```

```
#Filtering the transactions greater than 130000
Fltrd_Trans_sum
```

```
## # A tibble: 6 x 2
##   Country      sum
##   <chr>         <dbl>
## 1 Australia    137077.
## 2 EIRE         263277.
## 3 France       197404.
```

```
## 4 Germany      221698.
## 5 Netherlands  284662.
## 6 United Kingdom 8187806.
```

Question 4:

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

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

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

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

## Time difference of 8 days

```
# a:

#Percentage of number of transactions based on week days
Week_days_count = Online_Retail %>% group_by(Invoice_Day_Week) %>% summarise(percent = 100* n()/nrow(Online_Retail))

Week_days_count
```

```
## # A tibble: 6 x 2
##   Invoice_Day_Week percent
##   <chr>           <dbl>
## 1 Friday          15.2
## 2 Monday          17.6
## 3 Sunday          11.9
## 4 Thursday        19.2
## 5 Tuesday         18.8
## 6 Wednesday       17.5
```

```
# b:

#percentage of TransactionsValue

Week_days_sum = Online_Retail %>% group_by(Invoice_Day_Week) %>% summarise(sum=sum(TransactionValue))

#Calculating the percentage for TransactionValue by week days
Week_quan_pct = 100*(Week_days_sum$sum)/sum(Week_days_sum$sum)
```

```
#replacing the sum with the percentage value
```

```
Week_days_sum$sum = Week_quan_pct
```

```
Week_days_sum
```

```
## # A tibble: 6 x 2
```

```
##   Invoice_Day_Week    sum
```

```
##   <chr>           <dbl>
```

```
## 1 Friday          15.8
```

```
## 2 Monday          16.3
```

```
## 3 Sunday           8.27
```

```
## 4 Thursday        21.7
```

```
## 5 Tuesday         20.2
```

```
## 6 Wednesday       17.8
```

```
#c
```

```
#Percentage of TransactionsValue by month of the year
```

```
Invoice_month_sum = Online_Retail %>% group_by(New_Invoice_Month) %>% summarise(sum=sum(TransactionValue))
```

```
Month_quan_pct = 100*(Invoice_month_sum$sum)/sum(Invoice_month_sum$sum)
```

```
Invoice_month_sum$sum = Month_quan_pct
```

```
Invoice_month_sum
```

```
## # A tibble: 12 x 2
```

```
##   New_Invoice_Month    sum
```

```
##           <dbl> <dbl>
```

```
## 1             1  5.74
```

```
## 2             2  5.11
```

```
## 3             3  7.01
```

```
## 4             4  5.06
```

```
## 5             5  7.42
```

```
## 6             6  7.09
```

```
## 7             7  6.99
```

```
## 8             8  7.00
```

```
## 9             9 10.5
```

```
## 10            10 11.0
```

```
## 11            11 15.0
```

```
## 12            12 12.1
```

```
#d
```

```
#Filtering the Australia's transactions based on New_Invoice_date
```

```
Australia_trans = Online_Retail %>% filter(Country == "Australia") %>% group_by(New_Invoice_Date) %>% summarise(sum=sum(TransactionValue))
```

```
#Finding the date which has maximum number of transactions
```

```
Max_trans_date = Australia_trans[which.max(Australia_trans$total),]
```

```
Max_trans_date
```

```
## # A tibble: 1 x 2
##   New_Invoice_Date total
##   <date>             <int>
## 1 2011-06-15         139
```

```
#e
```

```
#Filtering the transactions for the hours between 7:00 to 20:00
```

```
Sum_quan = Online_Retail %>% filter( New_Invoice_Hour >=7) %>% group_by(New_Invoice_Hour) %>% summarise
```

```
#install.packages("zoo")
```

```
library(zoo)
```

```
##
```

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

```
## The following objects are masked from 'package:base':
```

```
##
```

```
##      as.Date, as.Date.numeric
```

```
#Adding the two consecutive rows
```

```
Consec_sum=rollapply(Sum_quan$sum_val,2,sum)
```

```
#Creating the maintainance column
```

```
maintainance=c(7:19)
```

```
#creating the dataframe for the maintainance and Consec_sum
```

```
Main_tab=data.frame(maintainance,Consec_sum)
```

```
#checking the minimum value of Consec_sum and the hour where they can start maintainance
```

```
maintainance_hour=Main_tab[which.min(Main_tab$Consec_sum),]
```

```
maintainance_hour
```

```
##      maintainance Consec_sum
```

```
## 13              19      40298
```

```
# From the above result,At 19th hour they can start maintainance
```

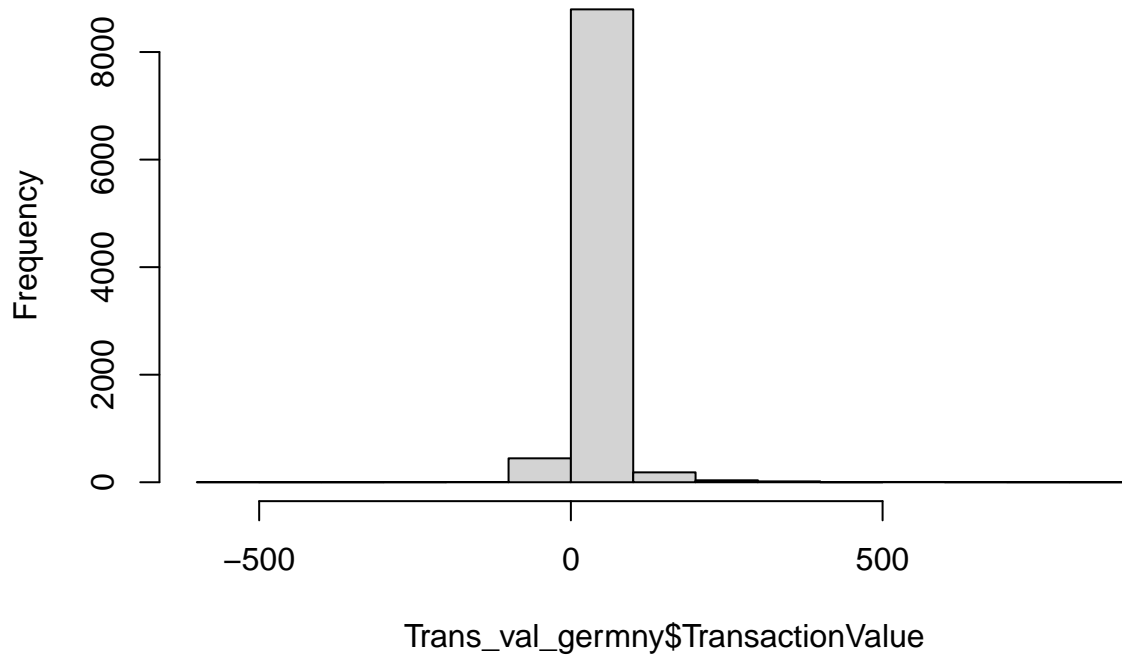
Question 5:

```
Trans_val_germny = filter(Online_Retail, Online_Retail$Country == "Germany")
```

```
#Plotting graph between transaction value with the frequency for Germany country
```

```
hist(Trans_val_germny$TransactionValue)
```

## Histogram of Trans\_val\_germny\$TransactionValue



Question 6:

```
#Removing the NA values of CustomerID Column
NA_OnlineRetail=Online_Retail[!is.na(Online_Retail$CustomerID),]

#Number of transactions with respect to CustomerID
Count_transactions = NA_OnlineRetail %>% group_by(CustomerID) %>% summarise(count=n())

#printing the row which has max count of transactions
Max_Count_transactions= Count_transactions[which.max(Count_transactions$count),]

# Adding the transaction value with respect to Customer ID
Sum_transactions = NA_OnlineRetail %>% group_by(CustomerID) %>% summarise(Numoftransactions=(sum(TransactionValue)))

#printing the row which has max sum of transaction value
Max_Sum_transactions= Sum_transactions[which.max(Sum_transactions$Numoftransactions),]

Max_Count_transactions

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

Max_Sum_transactions

## # A tibble: 1 x 2
```

```
## CustomerID Numoftransactions
## <int> <dbl>
## 1 14646 279489.
```

Question 7:

```
#Percentage of NA's for each column

NA_per = colMeans(is.na(Online_Retail))*100

NA_per
```

```
## 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
```

Question 8:

```
#Number of Transactions with missing customer ID

null_Customer = Online_Retail[is.na(Online_Retail$CustomerID),]

# Segregating the missing CustomerID based on countries

table(null_Customer$Country)
```

```
##
## Bahrain EIRE France Hong Kong Israel
## 2 711 66 288 47
## Portugal Switzerland United Kingdom Unspecified
## 39 125 133600 202
```

Question 10:

```
# Filtering the dataset for french customers

French_cstmrs = filter(Online_Retail, Country=="France" )

#Returnrate for the french customers

Return_rate = nrow(filter(French_cstmrs, Quantity<1))/nrow(French_cstmrs)

Return_rate
```

```
## [1] 0.01741264
```

Question 11:

```
#revenue of each product
Prd_revenue= Online_Retail %>% group_by(StockCode) %>% summarise(Sum_trnsvalue = sum(TransactionValue))

#Selecting the product with highest revenue
Prd_revenue[which.max(Prd_revenue$Sum_trnsvalue),]
```

```
## # A tibble: 1 x 2
##   StockCode Sum_trnsvalue
##   <chr>         <dbl>
## 1 DOT          206245.
```

Question 12:

```
#Number of unique customers
length(unique(Online_Retail$CustomerID))
```

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