Assignment_2_BA

Meghana udiga

2022-10-25

1.*how only countries accounting for more than* 1

creating a new variable name as transcation value

```
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
```

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

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

3.*showing the breakdown of transaction by countries*

```
    data <- summarise(group_by(Retail,Country),sum_1= sum(TransactionValue))
    Transaction <- filter(data,sum_1 >130000)
    Transaction
```

```
## # A tibble: 6 x 2
##   Country           sum_1
##   <chr>             <dbl>
## 1 Australia       137077.
## 2 EIRE            263277.
## 3 France          197404.
## 4 Germany         221698.
## 5 Netherlands     284662.
## 6 United Kingdom 8187806.
```

4.define the month as a separate numeric variable

```
library(zoo)
```

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

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

```
Temp=strptime(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"
```

```
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"))
#the  percentage  of  transactions by  numbers by  days  of  the  week
#(a)
a<-summarise(group_by(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) the  percentage  of  transactions by transaction volume by  days  of  the  week

b1<-summarise(group_by(Retail,Invoice_Day_Week),Transaction_Volume=sum(TransactionValue))
b2<-mutate(b1,percentage=(Transaction_Volume/sum(Transaction_Volume))*100)
b2
```

```
## # 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)the percentage of transactions (by transaction volume) by month of the year
c1<-summarise(group_by(Retail,New_Invoice_Month),Transaction_Volume=sum(TransactionValue))
c1<-mutate(c1,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
```

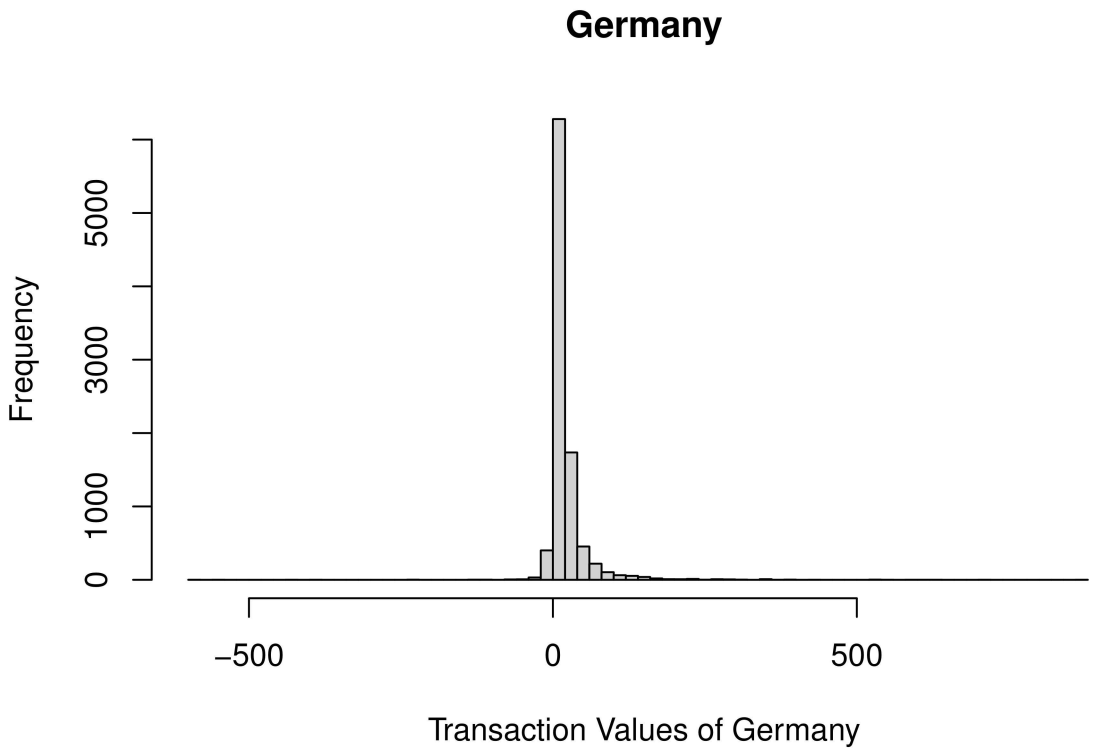
```
#(d) What was the date with the highest number of transactions from Australia?
Retail <- Retail %>% mutate(TransactionValue= Quantity * UnitPrice)

Retail %>% filter(Country == 'Australia') %>% group_by(New_Invoice_Date) %>% summarise(max=max(Transact
```

```
## # 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) hour of the day to start this so that the distribution is at minimum for the customers
e1<-summarise(group_by(Retail,New_Invoice_Hour),Transaction_min=n_distinct(InvoiceNo))
e1<-filter(e1,New_Invoice_Hour>=7&New_Invoice_Hour<=20)
e12<-rollapply(e1$Transaction_min,3,sum)
e123<-which.min(e12)
```

#5.plotting the histogram of the transaction value from germany

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




```
# 6. finding out the highest number of transactions and which customer is most valuable.
Retail1 <- na.omit(Retail)
result1 <- summarise(group_by(Retail1,CustomerID), sum2= sum(TransactionValue))
result1[which.max(result1$sum2),]
```

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

```
data2 <- table(Retail$CustomerID)
data2 <- as.data.frame(data2)
result2 <- data2[which.max(data2$Freq),]
result2
```

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

```
# 7.calculating the missing values of each variable in the dataset
    missing_values <- colMeans(is.na(Retail)*100)
    missing_values
```

```
##         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. the number of transactions with missing CustomerID records by countries?
    Retail2 <- Retail %>% filter(is.na(CustomerID)) %>% group_by(Country)
    summary(Retail2$Country)
```

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

```
#9 On average, how often the costumers comeback to the website for their next shopping?

Diff_Days <-  Retail %>% select(CustomerID,New_Invoice_Date) %>% group_by(CustomerID) %>% distinct(New_
Diff_Days
```

```
## # A tibble: 15,200 x 3
## # Groups:   CustomerID [2,992]
##    CustomerID New_Invoice_Date Days_Between
##         <int> <date>           <drtn>
##  1      18287 2011-10-12       143 days
##  2      18287 2011-10-28        16 days
##  3      18283 2011-01-23        17 days
##  4      18283 2011-02-28        36 days
##  5      18283 2011-04-21        52 days
##  6      18283 2011-05-23        32 days
##  7      18283 2011-06-14        22 days
##  8      18283 2011-06-23         9 days
##  9      18283 2011-07-14        21 days
## 10      18283 2011-09-05        53 days
## # ... with 15,190 more rows
```

```
mean(Diff_Days$Days_Between)
```

```
## Time difference of 38.4875 days
```

```
#10
Retail_table <- filter(Retail,Country=="France")
totalrow <- nrow(Retail_table)
cancel <- nrow(subset(Retail_table,TransactionValue<0))
cancel
```

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

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

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

```
TEST2=(cancel/8556)
TEST2
```

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

```
#11
Transaction_Value <- tapply(Retail$TransactionValue, Retail$StockCode  , sum)
Transaction_Value[which.max(Transaction_Value)]
```

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

```
# 12.unique customers present in the dataset are
unique_customers <- unique(Retail$CustomerID)

length(unique_customers)
```

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