Assignment_2

Rupesh Suragani

#Installing the libraries

```
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
```

Download the dataset, and use the read.csv() command to load the file into a R dataframe

```
Online_Retail <- read.csv("C:/Users/rupes/OneDrive/Desktop/Kent State University/BA/Assignment-2/Online_Retail.csv")

dim(Online_Retail)
```

```
## [1] 541909      8
```

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.

```
# Country wise transactions and its percentage which is more than 1%.
# Here the dataframe has to be grouped by country and to be summarized and filtered to more than 1%

Country_wise_Transactions <- Online_Retail %>%
  group_by(Country) %>%
  summarise(Transactions = n(), percentage = (n()/nrow(Online_Retail))*100) %>%
  filter(percentage > 1.0)

cat("Number of Transactions by Countries and their percentages (greater than 1%)", "\n")
```

```
## Number of Transactions by Countries and their percentages (greater than 1%)
```

```
Country_wise_Transactions
```

Country <chr>	Transactions <int>	percentage <dbl>
EIRE	8196	1.512431
France	8557	1.579047
Germany	9495	1.752139
United Kingdom	495478	91.431956

4 rows

2. Create a new variable 'TransactionValue' that is the product of the existing 'Quantity' and 'UnitPrice' variables. Add this variable to the dataframe.

```
# Creating the new variable "TransactionValue" to the dataframe (using Mutate function)

Online_Retail <- Online_Retail %>%
  mutate(TransactionValue = Quantity*UnitPrice)

dim(Online_Retail) #Total dimensions
```

```
## [1] 541909      9
```

```
colnames(Online_Retail)
```

```
## [1] "InvoiceNo"        "StockCode"        "Description"      "Quantity"
## [5] "InvoiceDate"      "UnitPrice"        "CustomerID"       "Country"
## [9] "TransactionValue"
```

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 by each country. Show this in total sum of transaction values. Show only countries with total transaction exceeding 130,000 British Pound.

```
# For showing only the countries with Total Transactions exceeding 130,000 pounds the data needs to be grouped country wise totaling each of their transaction values and filtering the values gr eater than 130,000.

Total_transaction_values <- Online_Retail %>% group_by(Country)%>%
  summarise(Total_TransValue = sum(TransactionValue)) %>%
  filter(Total_TransValue > 130000)

cat("Country wise total transaction values exceeding 130,000 pounds ", "\n")
```

```
## Country wise total transaction values exceeding 130,000 pounds
```

```
Total_transaction_values
```

Country <chr>	Total_TransValue <dbl>
Australia	137077.3
EIRE	263276.8
France	197403.9
Germany	221698.2
Netherlands	284661.5
United Kingdom	8187806.4

6 rows

4. 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. Click here for more information. First let’s convert ‘InvoiceDate’ into a POSIXlt object:

Check the variable using, head(Temp).

```
# Converting the 'InvoiceDate' to a POSIXlt object
Temp <- strptime(Online_Retail$InvoiceDate,format='%m/%d/%Y %H:%M',tz='GMT')

#This code uses the strptime function to convert the 'InvoiceDate' column in the 'Online_Retail'
data frame to a POSIXlt object. It specifies the format of the date and time in  your data, whic
h is 'month/day/year hour:minute', and sets the time zone to '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"
```

```
#Extracting the data component in a new column called New_Invoice_Date
Online_Retail$New_Invoice_Date <- as.Date(Temp)

#The Date objects have a lot of flexible functions. For example knowing two date values, the obj
ect allows you to know the difference between the two dates in terms of the number days.

# Calculating the difference between two dates
Online_Retail$New_Invoice_Date[20000] - Online_Retail$New_Invoice_Date[10]
```

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

```
#This line calculates the difference in days between the 20000th and 10th entries in the 'New_In
voice_Date' column.

#For the Hour, let's just take the hour (ignore the minute) and convert into a normal numerical
value:

# Extracting the hour component(Using as.numeric)
Online_Retail$New_Invoice_Hour <- as.numeric(format(Temp, "%H"))

# Extracting the hour component(Using weekdays)
Online_Retail$New_Invoice_Week <- weekdays(Online_Retail$New_Invoice_Date)

# Extracting the Month component(Using as.numeric)
Online_Retail$New_Invoice_Month <- as.numeric(format(Temp, "%m"))

#Now answer the flowing questions.
#a) Show the percentage of transactions (by numbers) by days of the week
# For this, the data needs to be grouped by New_Invoice_week column.
Online_Retail %>%
  group_by(New_Invoice_Week) %>%
  summarise(Trans_percentage = (n()/nrow(Online_Retail))*100)
```

New_Invoice_Week <chr>	Trans_percentage <dbl>
Friday	15.16731
Monday	17.55110
Sunday	11.87930
Thursday	19.16503
Tuesday	18.78692
Wednesday	17.45035

6 rows

```
#b) Show the percentage of transactions (by transaction volume) by days of the week
Online_Retail %>%
  group_by(New_Invoice_Week) %>%
  summarise(percent_by_transactions_volume = (sum(TransactionValue)/sum(Online_Retail$Transactio
nValue))*100)
```

New_Invoice_Week <chr>	percent_by_transactions_volume <dbl>
Friday	15.804787
Monday	16.297194
Sunday	8.265282

New_Invoice_Week <chr>	percent_by_transactions_volume <dbl>
Thursday	21.671867
Tuesday	20.170636
Wednesday	17.790232

6 rows

```
#c) Show the percentage of transactions (by transaction volume) by month of the year
Online_Retail %>%
  group_by(New_Invoice_Month) %>%
  summarise(percent_by_transactions_volume = (sum(TransactionValue)/sum(Online_Retail$Transactio
nValue))*100)
```

New_Invoice_Month <dbl>	percent_by_transactions_volume <dbl>
1	5.744919
2	5.109515
3	7.009487
4	5.059703
5	7.420519
6	7.090080
7	6.989308
8	7.003469
9	10.460751
10	10.984123

1-10 of 12 rows Previous **1** 2 Next

```
#d) What was the date with the highest number of transactions from Australia?
subset(Online_Retail, Country == "Australia") %>%
  group_by(New_Invoice_Date) %>%
  summarise(Transactions = n()) %>%
  arrange(desc(Transactions)) %>%
  head(1)
```

New_Invoice_Date <date>	Transactions <int>
2011-06-15	139

1 row

```
#e) The company needs to shut down the website for two consecutive hours for maintenance. What w
ould be the hour of the day to start this so that the distribution is at minimum for the custome
rs? The responsible IT team is available from 7:00 to 20:00 every day.
Online_Retail %>%
  group_by(New_Invoice_Hour) %>%
  summarise(percent_of_transactions = (n()/nrow(Online_Retail))*100) %>%
  arrange(percent_of_transactions)
```

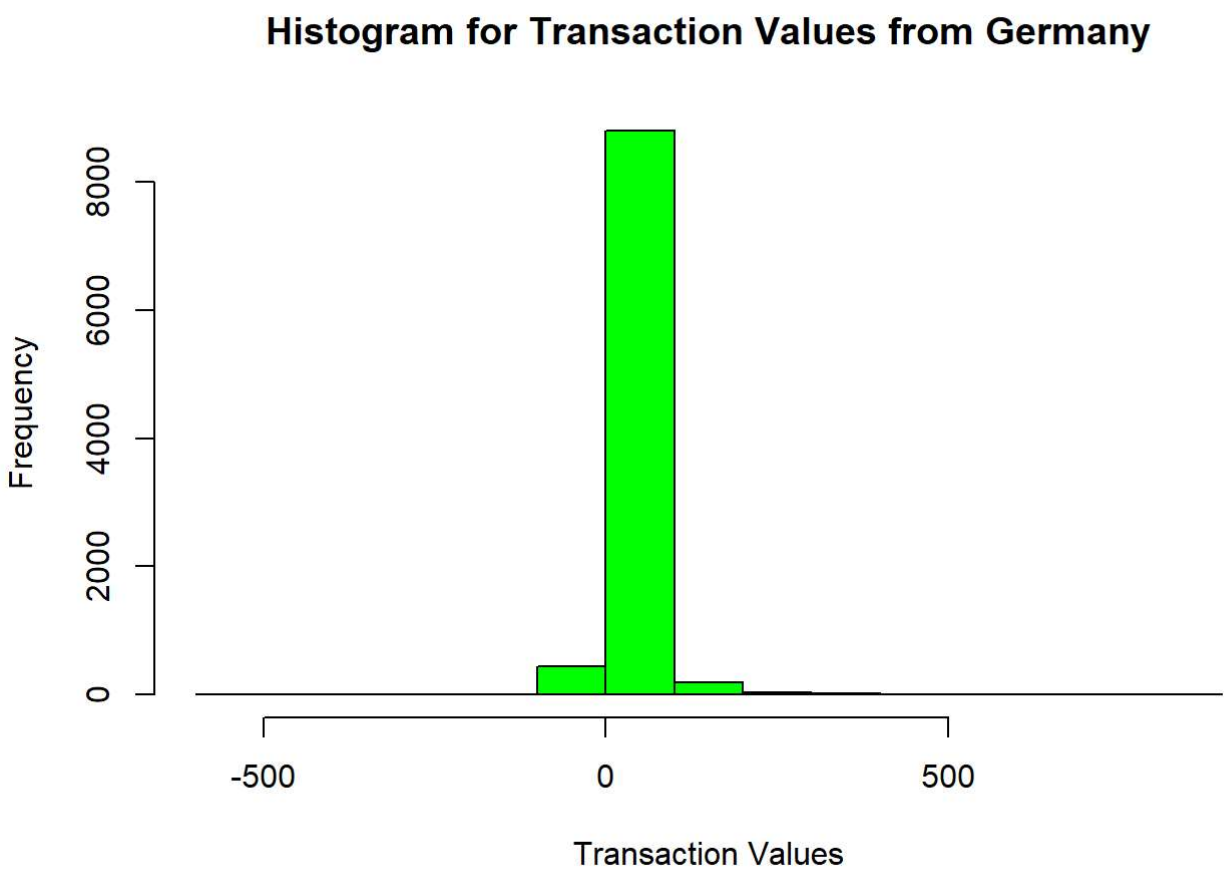
New_Invoice_Hour <dbl>	percent_of_transactions <dbl>
6	0.007565846
7	0.070676073
20	0.160728093
19	0.683694126
18	1.471464766
8	1.644002960
17	5.260846378
9	6.335381033
10	9.048936261
16	10.059991622

1-10 of 15 rows Previous **1** 2 Next

```
# Therefore, from running the above code, the best possible hour would be at 7.00 every day as t
he distribution (percent_of_transactions) at that time is minimum.
```

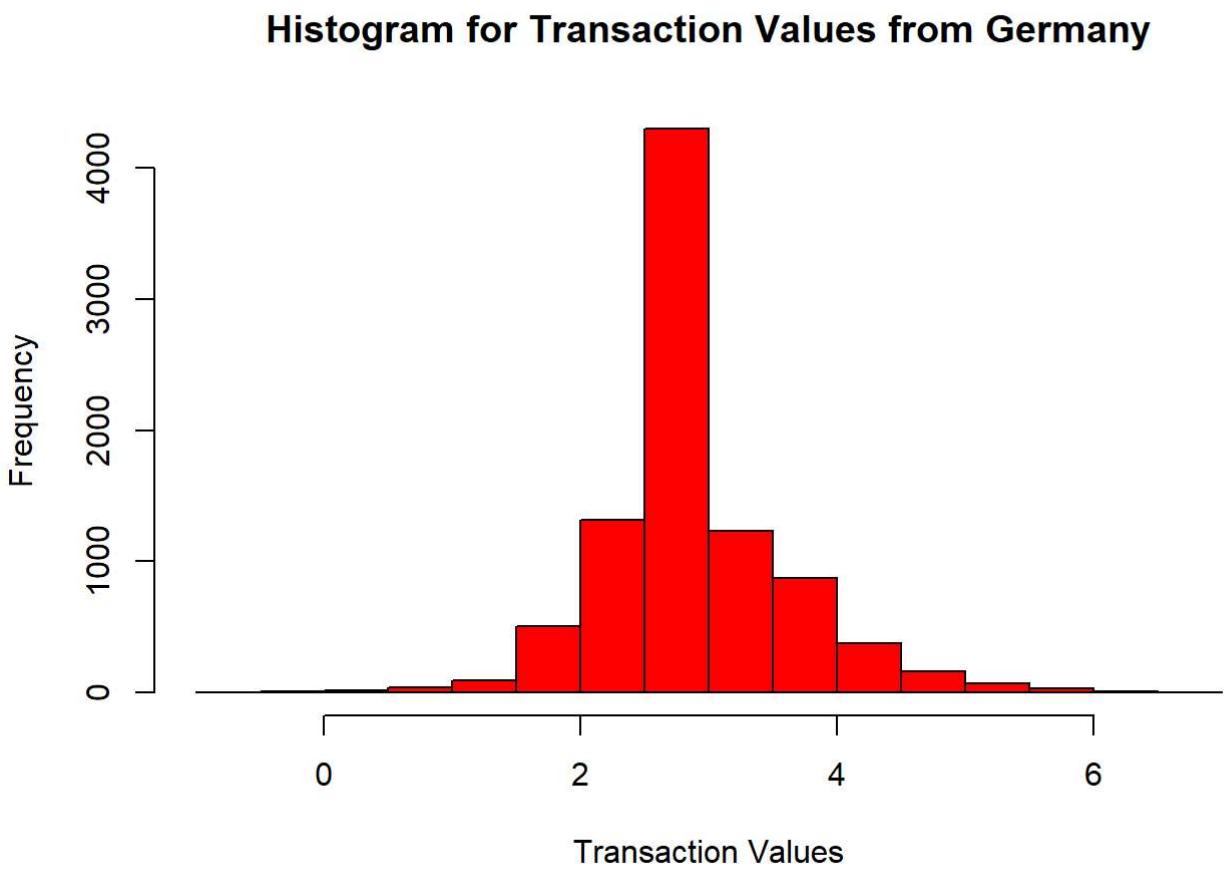
5. Plot the histogram of transaction values from Germany. Use the hist() function to plot.

```
# For plotting only the transaction values from Germany, the germany Transaction values data nee
ds to be taken out using the subset function to the dataframe.

#Filtering the data for germany
Germany_TransValues <- subset(Online_Retail, Country == "Germany")

# Plotting the data without scaling
hist(Germany_TransValues$TransactionValue,
     main = "Histogram for Transaction Values from Germany",
     xlab = "Transaction Values",
     ylab = "Frequency",
     col = "green")
```




```
#plotting the data with scaling using log
hist(log(Germany_TransValues$TransactionValue),
     main = "Histogram for Transaction Values from Germany",
     xlab = "Transaction Values",
     ylab = "Frequency",
     col = "red")
```

```
## Warning in log(Germany_TransValues$TransactionValue): NaNs produced
```



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

```
# Grouping the data by customerID to find the highest number of transactions
customers <- Online_Retail %>%
  na.omit %>% group_by(CustomerID) %>%
  summarise(Transactions = n()) %>%
  arrange(desc(Transactions))


# Grouping the data by customerID and adding there transaction values
Highest_customers <-  Online_Retail %>%
  na.omit %>% group_by(CustomerID) %>%
  summarise(total = sum(TransactionValue))%>%
  arrange(desc(total))

# print the data
cat("The customer with the highest number of transactions was:", as.numeric(customers[1,1]),
"\n")
```

```
## The customer with the highest number of transactions was: 17841
```

```
cat("The most valuble customer is:", as.numeric(Highest_customers[1,1]), "with the total transac
tion value of", as.numeric(Highest_customers[1,2]))
```

```
## The most valuble customer is: 14646 with the total transaction value of 279489
```

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

```
# get the count of missing values using is.na and Calculating the percentage of missing values f
or CustomerId as it only has missing values
missingValues <- colMeans(is.na(Online_Retail))

# Calculating the percentage of missing values for CustomerId as it only has missing values
Missing_percent <- missingValues["CustomerID"]*100

# print the data
cat("The Online_Retail dataframe consists of only CustomerID with missing values. The percentage
of data missing is:", Missing_percent)
```

```
## The Online_Retail dataframe consists of only CustomerID with missing values. The percentage o
f data missing is: 24.92669
```

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

```
Online_Retail %>%
  filter(is.na(CustomerID)) %>%    # Filter transactions with missing CustomerID records
  group_by(Country) %>%
  summarise(Missing_CustomerID = n())
```

Country <chr>	**Missing_CustomerID** <int>
Bahrain	2
EIRE	711
France	66
Hong Kong	288
Israel	47
Portugal	39
Switzerland	125
United Kingdom	133600
Unspecified	202

9 rows

9. On average, how often the costumers comeback to the website for their next shopping? (i.e. what is the average number of days between consecutive shopping)

```
# Sorting the data frame by CustomerID and New_Invoice_Date
Online_Retail <- Online_Retail[order(Online_Retail$CustomerID, Online_Retail$New_Invoice_Date),
]

# Calculating the average time difference in days
return_time <- mean(diff(Online_Retail$New_Invoice_Date), na.rm = TRUE)

# Display the average return time
cat("The average number of days that the customer comeback to their next shopping is ", as.numer
ic(return_time), "days")
```

```
## The average number of days that the customer comeback to their next shopping is  0.0005997328
days
```

10. In 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?

```
FrenchOrders <- subset(Online_Retail, Country == "France")
FrenchOrders_cancelled <- subset(Online_Retail, Country == "France" & Quantity <= 0)
France_ReturnRate <- (nrow(FrenchOrders_cancelled)/nrow(FrenchOrders))*100

cat("The return rate for the french Customers is",France_ReturnRate)
```

```
## The return rate for the french Customers is 1.741264
```

11. What is the product that has generated the highest revenue for the retailer? (i.e. item with the highest total sum of 'TransactionValue')

```
# Here, for generating the highest revenue by product the data needs to be grouped using stock c
ode.

Highest_revenue_product <- Online_Retail %>%
  group_by(Description) %>%
  summarise(TotalRevenue = sum(TransactionValue)) %>%
  arrange(desc(TotalRevenue))

cat("The product which has generated the highest revenue is ", as.character(Highest_revenue_prod
uct[1,1]), "and the total revenue is ", as.numeric(Highest_revenue_product[1,2]) )
```

```
## The product which has generated the highest revenue is  DOTCOM POSTAGE and the total revenue
is  206245.5
```

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

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

cat("The total no.of unique customers in the dataset is",Unique_customer )
```

```
## The total no.of unique customers in the dataset is 4373
```