MIS-64036 Assignment 1

Steven Spence

10/28/2018

MIS-64036: Business Analytics

Assignment I

Total Marks: 100  
Contribution to the Final Mark: 20% Submission deadline: 21 October Instructions: Please answer all questions. You should use R to solve the questions and include the screen shots in your submission. The Golden questions are optional and carries additional marks. This means that you will not lose marks if you do not answer that question. Please use the link provided on the Blackboard, under the assessment section, to upload your submissions. Late submissions, up to two days, are subject to 30% penalty. Submissions made more than two days after the deadline will not be graded.

Part A) Descriptive Statistics & Normal Distributions

1. a) What is the probability of obtaining a score greater than 700 on a GMAT test that has a mean of 494 and a standard deviation of 100? Assume GMAT scores are normally distributed (5 marks).

We will find the P(Score > 700) by utilizing the Z-score method:

Z-score: z = (x - μ) / σ

Z = (700 – 494) / 100

Z = 2.06

Using the standard normal probability table we find a z-score of 2.06 correlates to 0.9803 area of the curve to the left of the z-value. Therefore, there is a 98.03% probability of obtaining a score greater than 700.

b) What is the probability of getting a score between 350 and 450 on the same GMAT exam?(5 marks)

This problem will also utilize the Z-score method to determine the probability of a score between 350 and 450:

P(350 < Score < 450) = P(Score < 450) – P(Score < 350)

Z-score (Score < 450): z = (450 – 494) / 100

z = -0.44

P(Score < 450) = 0.3300 = 33.00%

Z-score (Score < 350): z = (350 – 494) / 100

z = -1.44

P(Score < 350) = 0.0749 = 7.49%

P(350 < Score < 450) = P(Score < 450) – P(Score < 350)

P(350 < Score < 450) = 33.00% - 7.49%

P(350 < Score < 450) = 25.51%

Therefore, there is a 25.51% probability of getting a GMAT score between 350 and 450 based on the given descriptive statistics.

1. Runzheimer International publishes business travel costs for various cities throughout the world. In particular, they publish per diem totals, which represent the average costs for the typical business traveler including three meals a day in business-class restaurants and single-rate lodging in business-class hotels and motels. If 86.65% of the per diem costs in Buenos Aires, Argentina, are less than $449 and if the standard deviation of per diem costs is $36, what is the average per diem cost in Buenos Aires? Assume that per diem costs are normally distributed (10 marks)

P(Costs < 449) = 86.65% = 0.8665

If we look up 0.8665 in the standard normal probabilities table, we will find this corresponds to a z-score of 1.11. Since we now know the z-score for a given value and standard deviation, we can use the z-score formula to find the average cost:

z = (x - μ) / σ

1.11 = (449 - μ) / 36

μ = 409.04

Based on the given information, we can conclude that the average per diem cost in Buenos Aires is $409.04.

1. Chris is interested in understanding the correlation between temperature in Kent, OH and Los Angeles, CA. He has got the following data for September 2017 from Alpha Knowledgebase. (5 marks)



He has sampled the mid-day temperature for days from Sep 2 to Sep 6 as follows:

Kent=c(59, 68, 78, 60)

Los\_Angeles=c(90, 82, 78, 75)

Calculate the correlation (Pearson Correlation Coefficient) between the temperatures of the two cities without using any R commands i.e. calculate step by step.

Pearson Correlation Coefficient Formula:

numerator = sum(((kent-mean(kent))\*(los\_angeles-mean(los\_angeles)))

numerator = -61.25

denominator = sqrt(sum((kent-mean(kent))^2)\* sqrt(sum((los\_angeles-mean(los\_angeles))^2)

denominator = 171.76

correlation = numerator / denominator

correlation = -0.357

Part B) Data Wrangling

For the questions in this part, you need to use the ‘Online Retail’ dataset which can be downloaded in CSV format from the course portal under the assignment folder. This is a transnational data set which contains all the transactions occurring between 01 Dec 2010 and 09 Dec 2011 for a UK-based and registered non-store online retail. The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers. The data contains the following attributes:

InvoiceNo: Invoice number. Nominal, a 6-digit integral number uniquely assigned to each transaction. If this code starts with letter ‘c’, it indicates a cancellation.

StockCode: Product (item) code. Nominal, a 5-digit integral number uniquely assigned to each distinct product.

Description: Product (item) name. Nominal.

Quantity: The quantities of each product (item) per transaction. Numeric.

InvoiceDate: Invoice Date and time. Numeric, the day and time when each transaction was generated.

UnitPrice: Unit price. Numeric, Product price per unit in sterling.

CustomerID: Customer number. Nominal, a 5-digit integral number uniquely assigned to each customer.

Country: Country name. Nominal, the name of the country where each customer resides.

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

# Loads "Online\_Retail" text file as a data frame named "Online\_Retail"  
  
Online\_Retail <- read.csv("/Users/stevespence/Desktop/M.S. Business Analytics/Fall - 2018/MIS-64036 Business Analytics/Assignments/Assignment 1/Online\_Retail.csv")

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. (5 marks)

# Load the dplyr package to execute following code  
  
library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.4

##   
## 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

# Groups data frame by country and then summarises transactions by count and percent  
# Filters out all countries that represent less than 1% of the transaction totals  
  
Online\_Retail %>%   
 group\_by(Country) %>%  
 summarise(n\_transactions = n(), percent\_total = 100\*(n()/nrow(Online\_Retail))) %>%  
 filter(percent\_total > 1.0) %>%   
 arrange(desc(percent\_total))

## Warning: package 'bindrcpp' was built under R version 3.4.4

## # A tibble: 4 x 3  
## Country n\_transactions percent\_total  
## <fct> <int> <dbl>  
## 1 United Kingdom 495478 91.4   
## 2 Germany 9495 1.75  
## 3 France 8557 1.58  
## 4 EIRE 8196 1.51

1. Create a new variable ‘TransactionValue’ that is the product of the exising ‘Quantity’ and ‘UnitPrice’ variables. Add this variable to the dataframe. (5 marks)

# Creates a new column titled "TransactionValue" and binds it to the original data frame  
# Head used to display first six rows of new data frame  
  
Online\_Retail <- cbind(Online\_Retail, TransactionValue = Online\_Retail$Quantity \* Online\_Retail$UnitPrice)  
head(Online\_Retail)

## InvoiceNo StockCode Description Quantity  
## 1 536365 85123A WHITE HANGING HEART T-LIGHT HOLDER 6  
## 2 536365 71053 WHITE METAL LANTERN 6  
## 3 536365 84406B CREAM CUPID HEARTS COAT HANGER 8  
## 4 536365 84029G KNITTED UNION FLAG HOT WATER BOTTLE 6  
## 5 536365 84029E RED WOOLLY HOTTIE WHITE HEART. 6  
## 6 536365 22752 SET 7 BABUSHKA NESTING BOXES 2  
## InvoiceDate UnitPrice CustomerID Country TransactionValue  
## 1 12/1/2010 8:26 2.55 17850 United Kingdom 15.30  
## 2 12/1/2010 8:26 3.39 17850 United Kingdom 20.34  
## 3 12/1/2010 8:26 2.75 17850 United Kingdom 22.00  
## 4 12/1/2010 8:26 3.39 17850 United Kingdom 20.34  
## 5 12/1/2010 8:26 3.39 17850 United Kingdom 20.34  
## 6 12/1/2010 8:26 7.65 17850 United Kingdom 15.30

1. 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. (10 marks)

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups transactions by country and then summarise it by sum of TransactionValue column  
# Filter out countries with spend less than 130,000 and arrange them in descending order  
  
Online\_Retail %>%   
 group\_by(Country) %>%  
 summarise(Total\_Spend = sum(TransactionValue)) %>%  
 filter(Total\_Spend > 130000) %>%   
 arrange(desc(Total\_Spend))

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

This is an optional question which carries additional marks (golden questions). 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:

Temp=strptime(Online\_Retail$InvoiceDate,format=‘%m/%d/%Y %H:%M’,tz=‘GMT’)

# Creates temporary variable that formats transaction date into mm/dd/yyyy format  
# Verify format by using the head command  
  
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"

Check the variable using, head(Temp). Now, let’s separate date, day of the week and hour components dataframe with names as New\_Invoice\_Date, Invoice\_Day\_Week and New\_Invoice\_Hour:

Online\_Retail$New\_Invoice\_Date <- as.Date(Temp)

# Formats the New\_Invoice\_Date column into a date format from the Temp variable  
  
Online\_Retail$New\_Invoice\_Date <- as.Date(Temp)

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

Online\_RetailNew\_Invoice\_Date[10]

# Example of how dates can be subtracted from each other and return the difference in values  
  
Online\_Retail$New\_Invoice\_Date[20000]- Online\_Retail$New\_Invoice\_Date[10]

## Time difference of 8 days

Also we can convert dates to days of the week. Let’s define a new variable for that

Online\_RetailNew\_Invoice\_Date)

# Convert dates to days of the week and assigns column title to Invoice\_Day\_Week  
  
Online\_Retail$Invoice\_Day\_Week= weekdays(Online\_Retail$New\_Invoice\_Date)

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

Online\_Retail$New\_Invoice\_Hour = as.numeric(format(Temp, “%H”))

# Create a new column with the transaction hour assigned to New\_Invoice\_Hour  
  
Online\_Retail$New\_Invoice\_Hour = as.numeric(format(Temp, "%H"))

Finally, lets define the month as a separate numeric variable too:

Online\_Retail$New\_Invoice\_Month = as.numeric(format(Temp, “%m”))

# Create a new column with the transaction month assigned to New\_Invoice\_Month  
  
Online\_Retail$New\_Invoice\_Month = as.numeric(format(Temp, "%m"))

Now answer the following questions:

1. Show the percentage of transactions (by numbers) by days of the week (extra 2 marks)

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups data frame by day of week, calculates the percentage of transactions (by number) by day, and returns values in decreasing order of percentage.  
  
Online\_Retail %>%  
 group\_by(Invoice\_Day\_Week) %>%  
 summarise(percent\_of\_transactions = 100\*(n()/nrow(Online\_Retail))) %>%  
 arrange(desc(percent\_of\_transactions))

## # A tibble: 6 x 2  
## Invoice\_Day\_Week percent\_of\_transactions  
## <chr> <dbl>  
## 1 Thursday 19.2  
## 2 Tuesday 18.8  
## 3 Monday 17.6  
## 4 Wednesday 17.5  
## 5 Friday 15.2  
## 6 Sunday 11.9

1. Show the percentage of transactions (by transaction volume) by days of the week (extra 1 marks)

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups data frame by day of week, calculates the percentage of transactions (by transaction value) by day, and returns values in decreasing order of percentage.  
  
Online\_Retail %>%  
 group\_by(Invoice\_Day\_Week) %>%  
 summarise(percent\_of\_transactions\_by\_volume = 100\*(sum(TransactionValue)/sum(Online\_Retail$TransactionValue))) %>%  
 arrange(desc(percent\_of\_transactions\_by\_volume))

## # A tibble: 6 x 2  
## Invoice\_Day\_Week percent\_of\_transactions\_by\_volume  
## <chr> <dbl>  
## 1 Thursday 21.7   
## 2 Tuesday 20.2   
## 3 Wednesday 17.8   
## 4 Monday 16.3   
## 5 Friday 15.8   
## 6 Sunday 8.27

1. Show the percentage of transactions (by transaction volume) by month of the year (extra 1 marks)

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups data frame by month of year, calculates the percentage of transactions (by transaction value) by month, and returns values in decreasing order of percentage.  
  
Online\_Retail %>%  
 group\_by(New\_Invoice\_Month) %>%  
 summarise(percent\_of\_transactions\_by\_volume = 100\*(sum(TransactionValue)/sum(Online\_Retail$TransactionValue))) %>%  
 arrange(desc(percent\_of\_transactions\_by\_volume))

## # A tibble: 12 x 2  
## New\_Invoice\_Month percent\_of\_transactions\_by\_volume  
## <dbl> <dbl>  
## 1 11 15.0   
## 2 12 12.1   
## 3 10 11.0   
## 4 9 10.5   
## 5 5 7.42  
## 6 6 7.09  
## 7 3 7.01  
## 8 8 7.00  
## 9 7 6.99  
## 10 1 5.74  
## 11 2 5.11  
## 12 4 5.06

1. What was the date with the highest number of transactions from Australia? (3 marks)

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
#Creates a subset of data for Australian transactions, groups by the date of invoice, and returns the top values for the year.  
  
subset(Online\_Retail, Country == "Australia") %>%  
 group\_by(New\_Invoice\_Date) %>%  
 summarise(n\_transactions = n()) %>%  
 top\_n(3)

## Selecting by n\_transactions

## # A tibble: 3 x 2  
## New\_Invoice\_Date n\_transactions  
## <date> <int>  
## 1 2011-06-15 139  
## 2 2011-07-19 137  
## 3 2011-08-18 97

1. 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? The responsible IT team is available from 7:00 to 20:00 every day(3 marks)

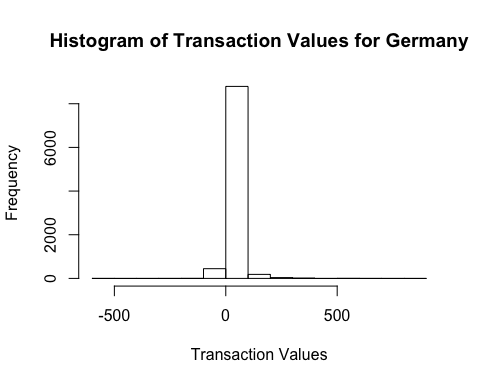
The best option would be to shut down the website from 6:00 – 8:00; however, this is not plausible with the I.T. department hours stated. Therefore, I would recommend having the I.T. department come in an hour early that day. If that is not possible, then the next best option would be to shut down the website from 18:00 – 20:00, which falls into the standard I.T. department hours.

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups data frame by hours for transactions, summarises data to return percent of transactions by number, and then returns values in ascending order.  
  
Online\_Retail %>%  
 group\_by(New\_Invoice\_Hour) %>%  
 summarise(percent\_of\_transactions = 100\*(n()/nrow(Online\_Retail))) %>%  
 arrange(percent\_of\_transactions)

## # A tibble: 15 x 2  
## New\_Invoice\_Hour percent\_of\_transactions  
## <dbl> <dbl>  
## 1 6 0.00757  
## 2 7 0.0707   
## 3 20 0.161   
## 4 19 0.684   
## 5 18 1.47   
## 6 8 1.64   
## 7 17 5.26   
## 8 9 6.34   
## 9 10 9.05   
## 10 16 10.1   
## 11 11 10.6   
## 12 14 12.5   
## 13 13 13.3   
## 14 15 14.3   
## 15 12 14.5

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

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Create new variable for Germany and plot the transaction values on a histogram  
  
Germany\_Transactions <- subset(Online\_Retail, Country == "Germany")  
hist(Germany\_Transactions$TransactionValue, main = "Histogram of Transaction Values for Germany", xlab = "Transaction Values", ylab = "Frequency")



1. Which customer had the highest number of transactions? Which customer is most valuable (i.e. highest total sum of transactions)? (10 marks)

CustomerID 17841 is the most valuable in terms of number of transactions, and CustomerID 14646 is most valuable in terms of value of transactions. However, this analysis shows that there are a significant number of transactions that are not being collected with a corresponding CustomerID (135,080 transactions to be precise). Therefore, it would be beneficial to try and get this information to these rows or implement a system that will prevent further transactions from missing values.

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups data by customer and then summarizes it based on count. Returns top three values displayed in descreasing value.   
  
Online\_Retail %>%  
 group\_by(CustomerID) %>%  
 summarise(n\_transactions = n()) %>%  
 top\_n(3) %>%  
 arrange(desc(n\_transactions))

## Selecting by n\_transactions

## # A tibble: 3 x 2  
## CustomerID n\_transactions  
## <int> <int>  
## 1 NA 135080  
## 2 17841 7983  
## 3 14911 5903

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups data by customer and then summarizes it based on transaction values. Returns top three values displayed in descreasing value.   
  
Online\_Retail %>%  
 group\_by(CustomerID) %>%  
 summarise(transaction\_sum = sum(TransactionValue)) %>%  
 top\_n(3) %>%  
 arrange(desc(transaction\_sum))

## Selecting by transaction\_sum

## # A tibble: 3 x 2  
## CustomerID transaction\_sum  
## <int> <dbl>  
## 1 NA 1447682.  
## 2 14646 279489.  
## 3 18102 256438.

1. Calculate the percentage of missing values for each variable in the dataset (5 marks). Hint colMeans():

CustomerID is the only column that is missing values. With this analysis, we see that 24.93% of the entries are missing a CustomerID value.

# Calculates the percentage of missing values for each variable in the data frame  
  
colMeans(is.na(Online\_Retail))

## InvoiceNo StockCode Description Quantity   
## 0.0000000 0.0000000 0.0000000 0.0000000   
## InvoiceDate UnitPrice CustomerID Country   
## 0.0000000 0.0000000 0.2492669 0.0000000   
## TransactionValue New\_Invoice\_Date Invoice\_Day\_Week New\_Invoice\_Hour   
## 0.0000000 0.0000000 0.0000000 0.0000000   
## New\_Invoice\_Month   
## 0.0000000

1. What are the number of transactions with missing CustomerID records by countries? (10 marks)

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Filter out values that are not NA, group by country, and summarise by total count  
  
Online\_Retail %>%  
 filter(is.na(Online\_Retail$CustomerID)) %>%  
 group\_by(Country) %>%  
 summarise(n\_missing\_ID = n()) %>%  
 arrange(desc(n\_missing\_ID))

## # A tibble: 9 x 2  
## Country n\_missing\_ID  
## <fct> <int>  
## 1 United Kingdom 133600  
## 2 EIRE 711  
## 3 Hong Kong 288  
## 4 Unspecified 202  
## 5 Switzerland 125  
## 6 France 66  
## 7 Israel 47  
## 8 Portugal 39  
## 9 Bahrain 2

1. 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) (Optional/Golden question: 18 additional marks!) Hint: 1. A close approximation is also acceptable and you may find diff() function useful.

According to this analysis, customers return to the website for their next shopping approximately every 78.42 days.

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Create a new data frame with all "NA" CustomerID's removed.  
  
Online\_Retail\_NA\_Removed <- na.omit(Online\_Retail)  
  
# Create a new data frame with cancelled transactions removed.  
  
Online\_Retail\_NA\_Neg\_Removed <- subset(Online\_Retail\_NA\_Removed, Quantity > 0)  
  
# Create a new data frame that only have customerID and transaction date  
  
Online\_Retail\_Subset <- Online\_Retail\_NA\_Neg\_Removed[,c("CustomerID","New\_Invoice\_Date")]  
  
# Create a new data frame to remove multiple invoices from same customer on same day  
  
Online\_Retail\_Subset\_Distinct <- distinct(Online\_Retail\_Subset)  
  
# Groups data set by CustomerID, arranges them by date, and finds the average time between consecutive transactions for each customer  
# Removes CustomerIDs that result in an NA value (i.e. only have one dinstinct transaction)  
# Summarises the data to find the average time between shopping trips for all CustomerIDs  
  
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  
## <time>   
## 1 78.42025 days

1. 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? (10 marks). Consider the cancelled transactions as those where the ‘Quantity’ variable has a negative value.

Return rate for French customers was found to be 1.74%

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
#Two new subsets created to calculate the total number of returns and total number of transactions for France. These are used to calculate the return rate.  
  
France\_Transactions\_Cancelled <- subset(Online\_Retail, Country == "France" & Quantity < 0)  
France\_Transactions <- subset(Online\_Retail, Country == "France")  
France\_Return\_Rate <- 100\*(nrow(France\_Transactions\_Cancelled) / nrow(France\_Transactions))  
France\_Return\_Rate

## [1] 1.741264

1. What is the product that has generated the highest revenue for the retailer? (i.e. item with the highest total sum of ‘TransactionValue’)(10 marks)

Dotcom postage was found to be the highest revenue for the retailer with a value of 206,245.

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
# Groups data by StockCode and item description and then summarizes it based on transaction values. Returns values displayed in descreasing value.   
  
Online\_Retail %>%  
 group\_by(StockCode, Description) %>%  
 summarise(transaction\_sum = sum(TransactionValue)) %>%  
 arrange(desc(transaction\_sum))

## # A tibble: 5,752 x 3  
## # Groups: StockCode [4,070]  
## StockCode Description transaction\_sum  
## <fct> <fct> <dbl>  
## 1 DOT DOTCOM POSTAGE 206245.  
## 2 22423 REGENCY CAKESTAND 3 TIER 164762.  
## 3 47566 PARTY BUNTING 98303.  
## 4 85123A WHITE HANGING HEART T-LIGHT HOLDER 97716.  
## 5 85099B JUMBO BAG RED RETROSPOT 92356.  
## 6 23084 RABBIT NIGHT LIGHT 66757.  
## 7 POST POSTAGE 66231.  
## 8 22086 "PAPER CHAIN KIT 50'S CHRISTMAS " 63792.  
## 9 84879 ASSORTED COLOUR BIRD ORNAMENT 58960.  
## 10 79321 CHILLI LIGHTS 53768.  
## # ... with 5,742 more rows

1. How many unique customers are represented in the dataset? You can use unique() and length() functions. (5 marks)

4,373 unique customers are found in this data set; however, there is a potential for significantly more unique customers if the NA values can be cleaned up with corresponding customerID values.

# Load the dplyr package to execute following code  
  
library(dplyr)  
  
#Returns the length of the CustomerID vector with duplicate entries removed.  
  
length(unique(Online\_Retail$CustomerID))

## [1] 4373

