COMP3162 Project 2

Phillip Llewellyn - 620118149, Annabelle Ellis - 620110644

3/30/2021

Project Part 02 – Teamwork

(2 persons per team)

[Weighted 15% of course marks]

DUE: April 07, 2021

1. Tweet Data Analysis [20]

a. Merge the tweets collected from project 01 by each person in the team. [1]

```
tweet_beer <- rbind(phillip_beer, subset(annabelle_beer, select = -c(X)) )
tweet_beverage <- rbind(phillip_beverage, subset(annabelle_beverage, select = -c(X)))
tweet_concert <- rbind(phillip_concert, subset(annabelle_concert, select = -c(X)))
tweet_party <- rbind(phillip_party, subset(annabelle_party, select = -c(X)))
```

- clean each dataframe again

```
# clean the text

tweet_beer.clean <- tweet_beer
tweet_beer.clean$text <- gsub("https.*","", tweet_beer.clean$text)
tweet_beer.clean$text <- gsub("http.*","", tweet_beer.clean$text)
tweet_beer.clean$text <- gsub("#.*","", tweet_beer.clean$text)
tweet_beer.clean$text <- gsub("@.*","", tweet_beer.clean$text)
tweet_beer.clean$text <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_beer.clean$text)
tweet_beer.clean$text <- gsub("U00..","", tweet_beer.clean$text)
tweet_beer.clean$text <- gsub("[^\x20-\x7E]","", tweet_beer.clean$text)

tweet_beverage.clean <- tweet_beverage
tweet_beverage.clean$text <- gsub("https.*","", tweet_beverage.clean$text)
tweet_beverage.clean$text <- gsub("http.*","", tweet_beverage.clean$text)
tweet_beverage.clean$text <- gsub("#.*","", tweet_beverage.clean$text)
tweet_beverage.clean$text <- gsub("@.*","", tweet_beverage.clean$text)
tweet_beverage.clean$text <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_beverage.clean$text)
tweet_beverage.clean$text <- gsub("U00..","", tweet_beverage.clean$text)
tweet_beverage.clean$text <- gsub("[^\x20-\x7E]","", tweet_beverage.clean$text)

tweet_concert.clean <- tweet_concert
```

```
tweet_concert.clean$text <- gsub("https.*","", tweet_concert.clean$text)
tweet_concert.clean$text <- gsub("http.*","", tweet_concert.clean$text)
tweet_concert.clean$text <- gsub("#.*","", tweet_concert.clean$text)
tweet_concert.clean$text <- gsub("@.*","", tweet_concert.clean$text)
tweet_concert.clean$text <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_concert.clean$text)
tweet_concert.clean$text <- gsub("U00..","", tweet_concert.clean$text)
tweet_concert.clean$text <- gsub("[^\x20-\x7E]","", tweet_concert.clean$text)

tweet_party.clean <- tweet_party
tweet_party.clean$text <- gsub("https.*","", tweet_party.clean$text)
tweet_party.clean$text <- gsub("http.*","", tweet_party.clean$text)
tweet_party.clean$text <- gsub("#.*","", tweet_party.clean$text)
tweet_party.clean$text <- gsub("@.*","", tweet_party.clean$text)
tweet_party.clean$text <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_party.clean$text)
tweet_party.clean$text <- gsub("U00..","", tweet_party.clean$text)
tweet_party.clean$text <- gsub("[^\x20-\x7E]","", tweet_party.clean$text)

# clean the locations

tweet_beer.clean$location <- gsub("https.*","", tweet_beer.clean$location)
tweet_beer.clean$location <- gsub("http.*","", tweet_beer.clean$location)
tweet_beer.clean$location <- gsub("#.*","", tweet_beer.clean$location)
tweet_beer.clean$location <- gsub("@.*","", tweet_beer.clean$location)
tweet_beer.clean$location <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_beer.clean$location)
tweet_beer.clean$location <- gsub("U00..","", tweet_beer.clean$location)
tweet_beer.clean$location <- gsub("[^\x20-\x7E]","", tweet_beer.clean$location)

tweet_beverage.clean$location <- gsub("https.*","", tweet_beverage.clean$location)
tweet_beverage.clean$location <- gsub("http.*","", tweet_beverage.clean$location)
tweet_beverage.clean$location <- gsub("#.*","", tweet_beverage.clean$location)
tweet_beverage.clean$location <- gsub("@.*","", tweet_beverage.clean$location)
tweet_beverage.clean$location <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_beverage.clean$location)
tweet_beverage.clean$location <- gsub("U00..","", tweet_beverage.clean$location)
tweet_beverage.clean$location <- gsub("[^\x20-\x7E]","", tweet_beverage.clean$location)

tweet_concert.clean$location <- gsub("https.*","", tweet_concert.clean$location)
tweet_concert.clean$location <- gsub("http.*","", tweet_concert.clean$location)
tweet_concert.clean$location <- gsub("#.*","", tweet_concert.clean$location)
tweet_concert.clean$location <- gsub("@.*","", tweet_concert.clean$location)
tweet_concert.clean$location <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_concert.clean$location)
tweet_concert.clean$location <- gsub("U00..","", tweet_concert.clean$location)
tweet_concert.clean$location <- gsub("[^\x20-\x7E]","", tweet_concert.clean$location)

tweet_party.clean$location <- gsub("https.*","", tweet_party.clean$location)
tweet_party.clean$location <- gsub("http.*","", tweet_party.clean$location)
tweet_party.clean$location <- gsub("#.*","", tweet_party.clean$location)
tweet_party.clean$location <- gsub("@.*","", tweet_party.clean$location)
tweet_party.clean$location <- gsub("[^[:alnum:][:blank:]?&/\\-]","", tweet_party.clean$location)
tweet_party.clean$location <- gsub("U00..","", tweet_party.clean$location)
tweet_party.clean$location <- gsub("[^\x20-\x7E]","", tweet_party.clean$location)
```

b. Remove all duplicate tweets in the newly merged set of tweets. A tweet is a duplicate if the text is exactly the same as the text in another tweet. In removing the duplicate tweets, it might be useful to

keep the one that has the highest retweet count. [2]

```
tweet_beer.clean.unique <- subset(tweet_beer.clean, !duplicated(tweet_beer.clean$text))
tweet_beverage.clean.unique <- subset(tweet_beverage.clean, !duplicated(tweet_beverage.clean$text))
tweet_concert.clean.unique <- subset(tweet_concert.clean, !duplicated(tweet_concert.clean$text))
tweet_party.clean.unique <- subset(tweet_party.clean, !duplicated(tweet_party.clean$text))
```

c. Explore the merged tweets and provide descriptive statistics. [3]

- Beer descriptive statistics

```
beer_count = nrow(tweet_beer.clean.unique)

beer_user_most_retweet = unname( tweet_beer.clean.unique[
                max(tweet_beer.clean.unique$followers_count)==tweet_beer.clean.unique$followers_count
                ,"screen_name"
                ]
              )

beer_most_retweet = unname(
              tweet_beer.clean.unique[
                max(tweet_beer.clean.unique$retweet_count)==tweet_beer.clean.unique$retweet_count
                ,"text"
                ]
              )

beer_location_count <- table(tweet_beer.clean.unique$location)
beer_location_count <- as.data.frame(beer_location_count)

# remove empty location row
beer_location_count <- subset(beer_location_count, beer_location_count$Var1 != "")


beer_location_most_tweet = unname(
              beer_location_count[
                max(beer_location_count$Freq)==beer_location_count$Freq
                ,"Var1"
                ]
              )
```

```
## [1] "--------------------------------------------------"

## [1] "                    Beer"

## [1] "--------------------------------------------------"

## [1] "The number of Twets retrieved are:  5667"

## [1] "The user with the most followers:  ABC"

## [1] "The tweets with the most retweet:  its the weekend baby youknow what that means  its time to dr
```

```
## [1] "The location with the most tweets:  Los Angeles CA"
```

- Beverage descriptive statistics

```
beverage_count = nrow(tweet_beverage.clean.unique)

beverage_user_most_retweet = unname( tweet_beverage.clean.unique[
                max(tweet_beverage.clean.unique$followers_count)==tweet_beverage.clean.unique$followers
                ,"screen_name"
                ]
              )

beverage_most_retweet = unname(
              tweet_beverage.clean.unique[
                max(tweet_beverage.clean.unique$retweet_count)==tweet_beverage.clean.unique$retweet_cou
                ,"text"
                ]
              )

beverage_location_count <- table(tweet_beverage.clean.unique$location)
beverage_location_count <- as.data.frame(beverage_location_count)

# remove empty location row
beverage_location_count <- subset(beverage_location_count, beverage_location_count$Var1 != "")


beverage_location_most_tweet = unname(
              beverage_location_count[
                max(beverage_location_count$Freq)==beverage_location_count$Freq
                ,"Var1"
                ]
              )
```

```
## [1] "-------------------------------------------------"

## [1] "                    beverage"

## [1] "-------------------------------------------------"

## [1] "The number of Twets retrieved are:  12863"

## [1] "The user with the most followers:  timesofindia"

## [1] "The tweets with the most retweet:  Erica Nlewedim bags new endorsement deal with beverage compa

## [1] "The location with the most tweets:  United States"
```

- party descriptive statistics

```
party_count = nrow(tweet_party.clean.unique)

party_user_most_retweet = unname( tweet_party.clean.unique[
                max(tweet_party.clean.unique$followers_count)==tweet_party.clean.unique$followers_count
                ,"screen_name"
                ]
              )

party_most_retweet = unname(
              tweet_party.clean.unique[
                max(tweet_party.clean.unique$retweet_count)==tweet_party.clean.unique$retweet_count
                ,"text"
                ]
              )

party_location_count <- table(tweet_party.clean.unique$location)
party_location_count <- as.data.frame(party_location_count)

# remove empty location row
party_location_count <- subset(party_location_count, party_location_count$Var1 != "")


party_location_most_tweet = unname(
              party_location_count[
                max(party_location_count$Freq)==party_location_count$Freq
                ,"Var1"
                ]
              )
```

```
## [1] "-------------------------------------------------"

## [1] "                    party"

## [1] "-------------------------------------------------"

## [1] "The number of Twets retrieved are:  11297"

## [1] "The user with the most followers:  XHNews"

## [1] "The tweets with the most retweet:  Family welcome Mark back home party F389 "

## [1] "The location with the most tweets:  United States"
```

- concert descriptive statistics

```
concert_count = nrow(tweet_concert.clean.unique)

concert_user_most_retweet = unname( tweet_concert.clean.unique[
                max(tweet_concert.clean.unique$followers_count)==tweet_concert.clean.unique$followers_c
                ,"screen_name"
                ]
```

```
              )

concert_most_retweet = unname(
              tweet_concert.clean.unique[
                max(tweet_concert.clean.unique$retweet_count)==tweet_concert.clean.unique$retweet_count
                ,"text"
                ]
              )

concert_location_count <- table(tweet_concert.clean.unique$location)
concert_location_count <- as.data.frame(concert_location_count)

# remove empty location row
concert_location_count <- subset(concert_location_count, concert_location_count$Var1 != "")


concert_location_most_tweet = unname(
              concert_location_count[
                max(concert_location_count$Freq)==concert_location_count$Freq
                ,"Var1"
                ]
              )
```

```
## [1] "--------------------------------------------------"

## [1] "                    concert"

## [1] "--------------------------------------------------"

## [1] "The number of Twets retrieved are:  9802"

## [1] "The user with the most followers:  cnni"

## [1] "The tweets with the most retweet:  keep wearing your mask so we can scream what a shame shes fu

## [1] "The location with the most tweets:  she/her"
```

Analysis on emotions

```
t.beer <- tweet_beer.clean.unique
t.beverage <- tweet_beverage.clean.unique
t.party <- tweet_party.clean.unique
t.concert <- tweet_concert.clean.unique

#extract emotions
t.beer.emot <- get_nrc_sentiment(t.beer$text)
t.beverage.emot <- get_nrc_sentiment(t.beverage$text)
t.party.emot <- get_nrc_sentiment(t.party$text)
t.concert.emot <- get_nrc_sentiment(t.concert$text)
```

```
t.beer <- cbind(t.beer, t.beer.emot)
t.beverage <- cbind(t.beverage, t.beverage.emot)
t.party <- cbind(t.party, t.party.emot)
t.concert <- cbind(t.concert, t.concert.emot)

#extract sentiments
t.beer.sent <-  get_sentiment(t.beer$text)
t.beverage.sent <-  get_sentiment(t.beverage$text)
t.party.sent <-  get_sentiment(t.party$text)
t.concert.sent <-  get_sentiment(t.concert$text)

t.beer <- cbind(t.beer, t.beer.sent)
t.beverage <- cbind(t.beverage, t.beverage.sent)
t.party <- cbind(t.party, t.party.sent)
t.concert <- cbind(t.concert, t.concert.sent)
```

d. What are the dominant emotions associated with beverages in any two locations? [4]

```
# The two location that were chosen is the two top locations for beverages
us_data <- subset(t.beverage, t.beverage$location=="United States",  select = names(t.beer.emot) )
us_data.sum <- colSums(us_data)

ny_data <- subset(t.beverage, t.beverage$location=="New York, NY",  select = names(t.beer.emot) )
ny_data.sum <- colSums(ny_data)

# get the dominant emotions

us_dom_emot <- names(us_data.sum)[match(max(us_data.sum),us_data.sum)]

ny_dom_emot = names(ny_data.sum)[match(max(ny_data.sum),ny_data.sum)]
```

```
## [1] "The dominant emotions are: "

## [1] "United States :  positive"

## [1] "New York, NY :  anger"
```

e. What are the dominant emotions in the overall dataset? [2]

```
t.beer.emot.sum <- colSums(t.beer.emot)
t.beer.emot.dom <- names(t.beer.emot.sum)[match(max(t.beer.emot.sum),t.beer.emot.sum)]

t.beverage.emot.sum <- colSums(t.beverage.emot)
t.beverage.emot.dom <- names(t.beverage.emot.sum)[match(max(t.beverage.emot.sum),t.beverage.emot.sum)]

t.party.emot.sum <- colSums(t.party.emot)
t.party.emot.dom <- names(t.party.emot.sum)[match(max(t.party.emot.sum),t.party.emot.sum)]

t.concert.emot.sum <- colSums(t.concert.emot)
t.concert.emot.dom <- names(t.concert.emot.sum)[match(max(t.concert.emot.sum),t.concert.emot.sum)]
```

```
## [1] "The dominant emotions in the oeral data set are: "
```

```
## [1] "Beers :  positive"
```

```
## [1] "Beverages :  positive"
```

```
## [1] "Parties :  positive"
```

```
## [1] "Concerts :  positive"
```

f. What is the overall sentiment in tweets regarding "beverages" and "party or concert" (separately)?[4]

```
beer_overall_sent <- sum(t.beer.sent)
bev_overall_sent <- sum(t.beverage.sent)
party_overall_sent <- sum(t.party.sent)
concert_overall_sent <- sum(t.concert.sent)
```

```
## [1] "The overall sentiments are: "
```

```
## [1] "Beer :  1674.05 ,  Postive"
```

```
## [1] "Beverage :  9222.8 ,  Postive"
```

```
## [1] "Party :  6755.95 ,  Postive"
```

```
## [1] "Concert :  3769.85 ,  Postive"
```

g. Conduct ONE additional analysis of your choice to discover any further useful insights.[4]

```
# beer

beer_location_sent <- beer_location_count
beer_location_sent$sentiment <- c(0)
cnt <- NROW(beer_location_sent$Var1)

for(i in 1:cnt){
  data_sub <- subset(t.beer, t.beer$location == toString(beer_location_sent$Var1[i]))
  beer_location_sent$sentiment[i] <- sum(data_sub$t.beer.sent)
}

beer_loc_most_pos_sent <- beer_location_sent$Var1[match(max(beer_location_sent$sentiment),beer_location

# beverage

beverage_location_sent <- beverage_location_count
beverage_location_sent$sentiment <- c(0)
cnt <- NROW(beverage_location_sent$Var1)

for(i in 1:cnt){
  data_sub <- subset(t.beverage, t.beverage$location == toString(beverage_location_sent$Var1[i]))
  beverage_location_sent$sentiment[i] <- sum(data_sub$t.beverage.sent)
}
```

```
beverage_loc_most_pos_sent <- beverage_location_sent$Var1[match(max(beverage_location_sent$sentiment),be

# party

party_location_sent <- party_location_count
party_location_sent$sentiment <- c(0)
cnt <- NROW(party_location_sent$Var1)

for(i in 1:cnt){
  data_sub <- subset(t.party, t.party$location == toString(party_location_sent$Var1[i]))
  party_location_sent$sentiment[i] <- sum(data_sub$t.party.sent)
}

party_loc_most_pos_sent <- party_location_sent$Var1[match(max(party_location_sent$sentiment),party_loca

# concert

concert_location_sent <- concert_location_count
concert_location_sent$sentiment <- c(0)
cnt <- NROW(concert_location_sent$Var1)

for(i in 1:cnt){
  data_sub <- subset(t.concert, t.concert$location == toString(concert_location_sent$Var1[i]))
  concert_location_sent$sentiment[i] <- sum(data_sub$t.concert.sent)
}

concert_loc_most_pos_sent <- concert_location_sent$Var1[match(max(concert_location_sent$sentiment),conc
```

```
## [1] "The location with the most positive sentiment are: "
```

```
## [1] "Beer :  Atlanta GA"
```

```
## [1] "Beverage :  United States"
```

```
## [1] "Party :  London England"
```

```
## [1] "Concert :  Chicago IL"
```

2. Collect, Explore, Prepare Structured Data [20 marks]

a. Download the datafile consumer_pt02_2021.csv from OurVLE

```
consumer_data <- read.csv("consumer_pt02_2021.csv", stringsAsFactors = TRUE)
```

b. Explore the data and provide details on all fields retrieved. You should ensure all features in the dataset (each column) are reviewed and summarized to verify things such as value ranges, missing values etc. Be sure to generate relevant graphical representations where necessary to demonstrate your review and decision making. [7]

```r
# clean the data
consumer_data.clean <- consumer_data
consumer_data.clean$X <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$X)
consumer_data.clean$Region <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Region)
consumer_data.clean$Country <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Country)
consumer_data.clean$Sales.Channel <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Sales.Cl
consumer_data.clean$Order.Priority <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Order.P
consumer_data.clean$Order.Date <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Order.Date)
consumer_data.clean$Order.ID <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Order.ID)
consumer_data.clean$Ship.Date <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Ship.Date)
consumer_data.clean$Units.Sold <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Units.Sold)
consumer_data.clean$Unit.Price <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Unit.Price)
consumer_data.clean$Unit.Price <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Unit.Price)
consumer_data.clean$Unit.Cost <- gsub(",","", consumer_data.clean$Unit.Cost)
consumer_data.clean$Total.Revenue <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Total.Re
consumer_data.clean$Total.Cost <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Total.Cost)
consumer_data.clean$Total.Profit <- gsub("[^[:alnum:][:blank:]?&/\\-]","", consumer_data.clean$Total.Pro

consumer_data.clean$Order.Date <- as.Date(consumer_data.clean$Order.Date, "%m/%d/%Y")
consumer_data.clean$Ship.Date <- as.Date(consumer_data.clean$Ship.Date, "%m/%d/%Y")



consumer_data.clean$Units.Sold <- suppressWarnings(as.numeric(consumer_data.clean$Units.Sold))
consumer_data.clean$Unit.Price <- suppressWarnings(as.numeric(consumer_data.clean$Unit.Price))
consumer_data.clean$Unit.Cost <- suppressWarnings(as.numeric(consumer_data.clean$Unit.Cost))
consumer_data.clean$Total.Revenue <- suppressWarnings(as.numeric(consumer_data.clean$Total.Revenue))
consumer_data.clean$Total.Cost <- suppressWarnings(as.numeric(consumer_data.clean$Total.Cost))
consumer_data.clean$Total.Profit <- suppressWarnings(as.numeric(consumer_data.clean$Total.Profit))


#summary(consumer_data.clean)



#summary(consumer_data.clean$Sales.Channel)

#count(consumer_data.clean, Sales.Channel)

# show the summary of each of the numerical data
summary(consumer_data.clean$Units.Sold)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##       1    2506    5008    5002    7496   10000     618
```

```r
summary(consumer_data.clean$Unit.Price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     933    4372   15258   21818   42189   66827     943
```

```r
summary(consumer_data.clean$Unit.Cost)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    6.92   56.67  117.11  187.98  364.69  524.96     618
```

```
summary(consumer_data.clean$Total.Revenue)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##       933   7224424  33602436  96013660 114205623 668069519       879
```

```
summary(consumer_data.clean$Total.Cost)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##       519   8837955  33836479  80589495  97606706 524907504       618
```

```
summary(consumer_data.clean$Total.Profit)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's
##       241   2466464  13679728  30861337  48474956 173817839      1572
```

```
# show the number of empty strings in the data set
apply(consumer_data.clean,2,function(c) sum(c==""))
```

```
##         Region        Country              X  Sales.Channel Order.Priority
##           1080              0              2             85              0
##     Order.Date       Order.ID      Ship.Date     Units.Sold     Unit.Price
##             NA              0             NA             NA             NA
##      Unit.Cost  Total.Revenue     Total.Cost   Total.Profit
##             NA             NA             NA             NA
```

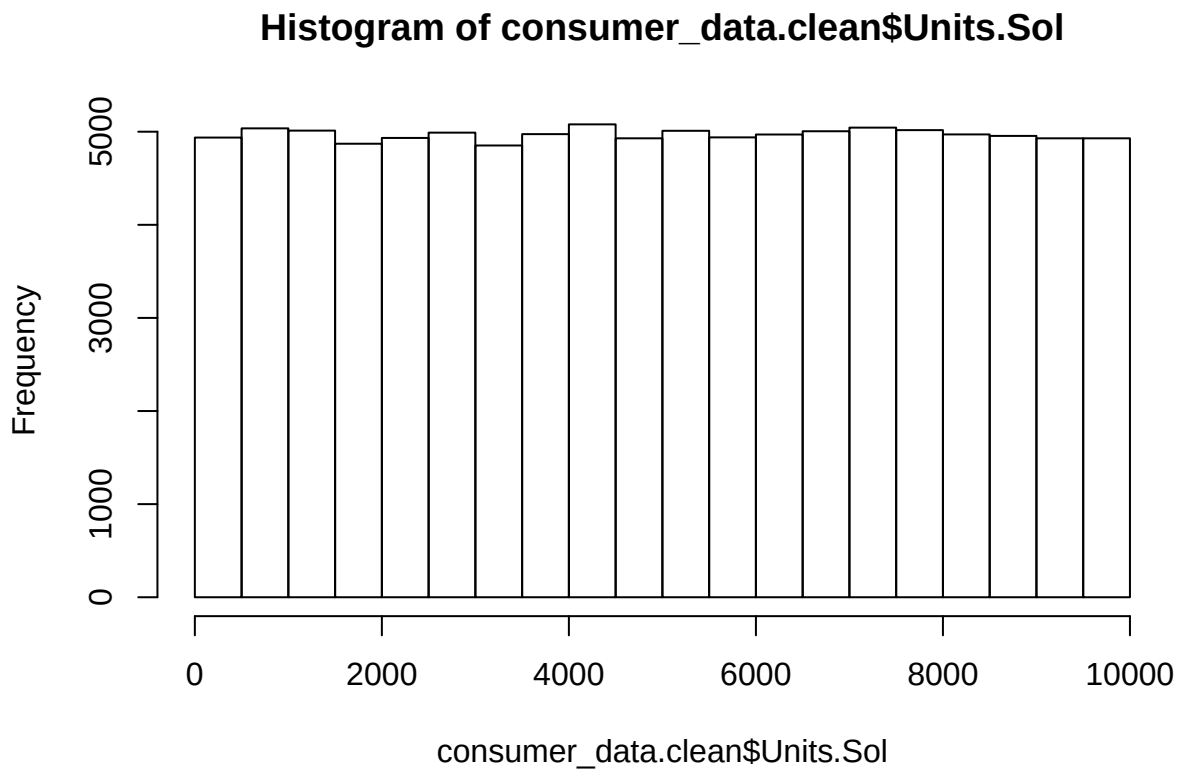
```
# show the number of na values in the dataset for each column
apply(consumer_data.clean,2,function(c) sum(is.na(c)))
```

```
##         Region        Country              X  Sales.Channel Order.Priority
##              0              0              0              0              0
##     Order.Date       Order.ID      Ship.Date     Units.Sold     Unit.Price
##            693              0            702            618            943
##      Unit.Cost  Total.Revenue     Total.Cost   Total.Profit
##            618            879            618           1572
```

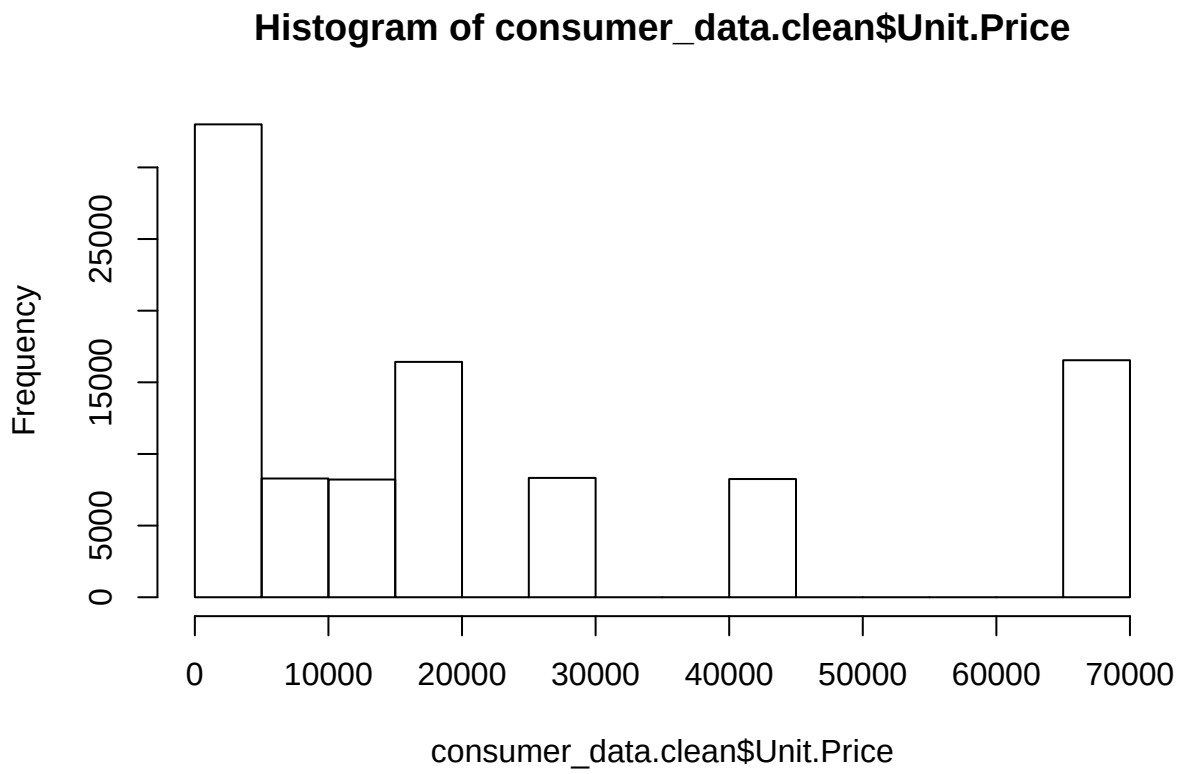
```
summary(consumer_data.clean$Order.Date)
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2012-01-01" "2013-11-25" "2015-10-14" "2015-10-15" "2017-09-06" "2019-07-28"
##         NA's
##        "693"
```

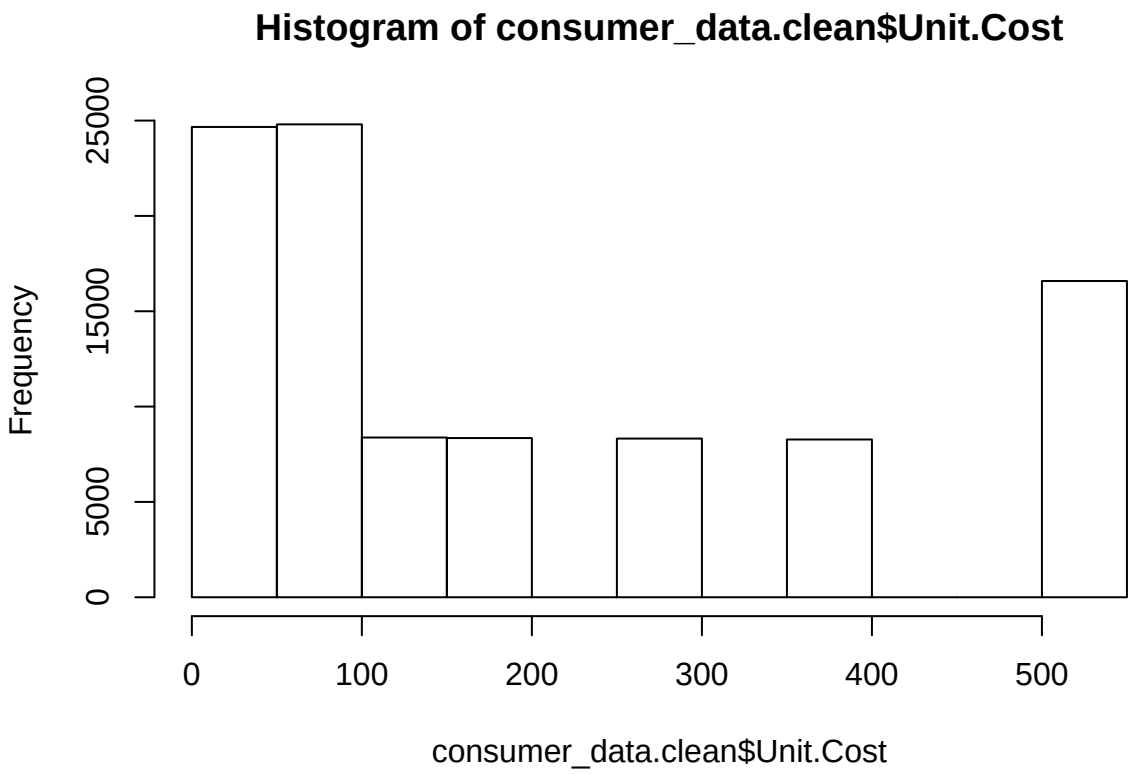
```
hist(consumer_data.clean$Units.Sol)
```



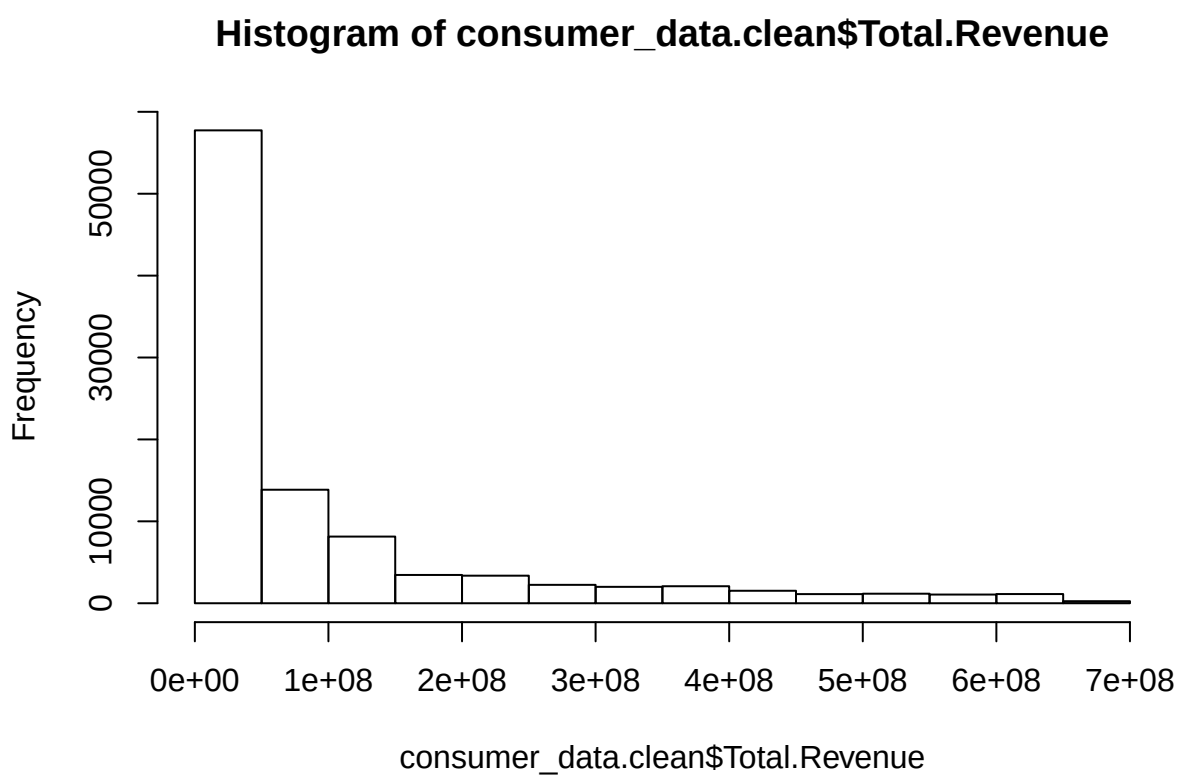

```
hist(consumer_data.clean$Unit.Price)
```



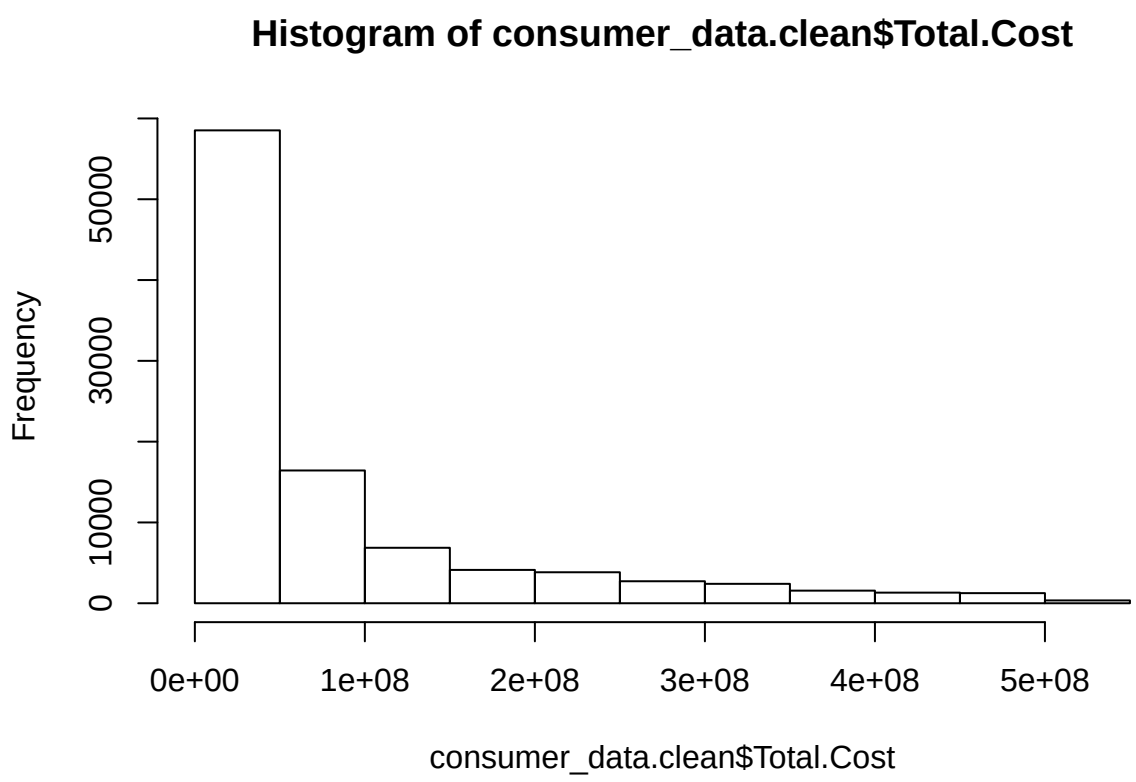

```
hist(consumer_data.clean$Unit.Cost)
```



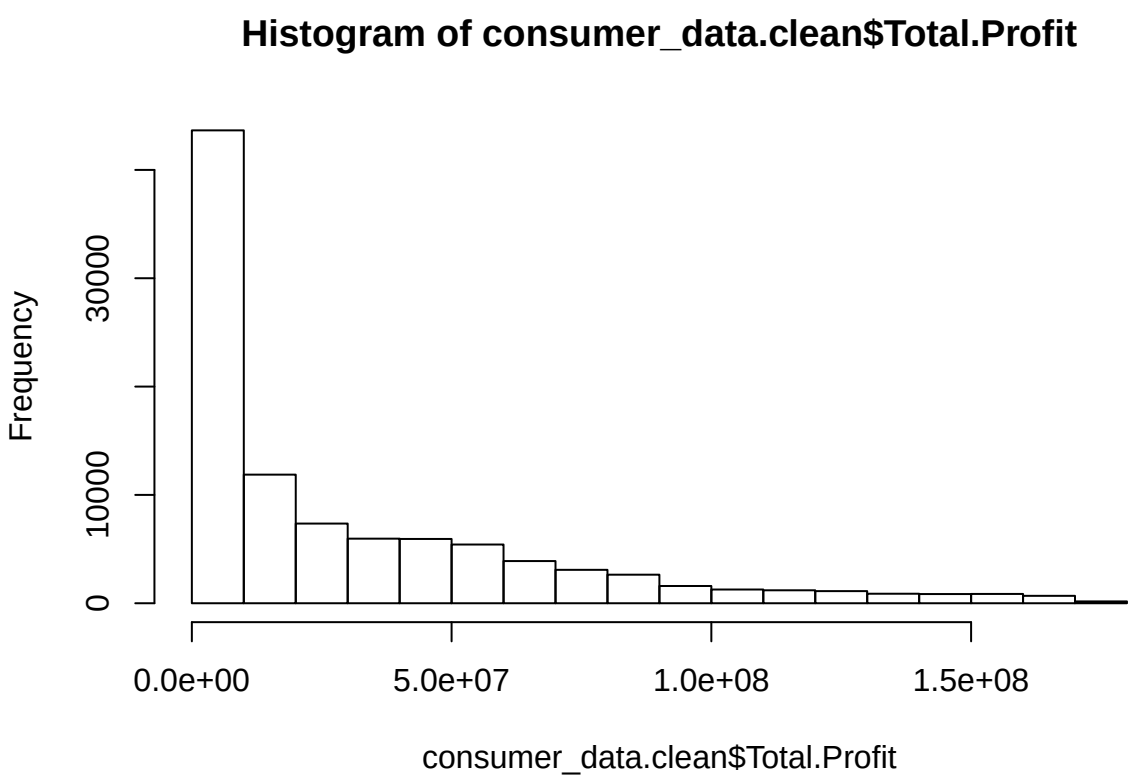

```
hist(consumer_data.clean$Total.Revenue)
```




```
hist(consumer_data.clean$Total.Cost)
```




```
hist(consumer_data.clean$Total.Profit)
```



c. Fix noise, outlier and any other issues discovered (example: na values). You must provide discussion / explanation of all activities done and why each decision has been made. [8]

The analysis above is able to show us that the regions, sales channel and X holds qualitative data that has missing values. The data for the region which are missing or having a question mark were replaced with unknown as the region. This was done because there was little information that could be used to help to decide on the correct value. The missing sales channel was set to unknown due to insufficient data to set a value, however for values of Yes and k the chaneel was set to online. This was done due to an assumption that was made that the user was affirming the use of the online channel. The numeric data were calculated values. Therefore, an arithmatic calculation was used to correct the possible correctable missing values. The values that failed to be calculated the mean value was uded to replace the NA. This was done because for each column the values are not far apart from the mean value.

```
#replace the column empty data

consumer_data.clean[consumer_data.clean$Region == '',"Region"] <- "Unknown"
consumer_data.clean[consumer_data.clean$Region == '?',"Region"] <- "Unknown"
consumer_data.clean[consumer_data.clean$X == '',"X"] <- "Unknown"

consumer_data.clean[consumer_data.clean$Sales.Channel == '',"Sales.Channel"] <- "Unknown"
consumer_data.clean[consumer_data.clean$Sales.Channel == 'k',"Sales.Channel"] <- "Online"
consumer_data.clean[consumer_data.clean$Sales.Channel == 'YES',"Sales.Channel"] <- "Online"

# replace the NA values with the calculated answer
```

```r
for(i in 1:NROW( consumer_data.clean$Units.Sold )){
  if( is.na(consumer_data.clean$Units.Sold[i]) == TRUE){
    consumer_data.clean$Units.Sold[i] = consumer_data.clean$Total.Revenue[i] / consumer_data.clean$Unit
  }
}

for(i in 1:NROW( consumer_data.clean$Unit.Price )){
  if( is.na(consumer_data.clean$Unit.Price[i]) ){
     consumer_data.clean$Unit.Price[i] <- consumer_data.clean$Total.Revenue[i] / consumer_data.clean$Uni
  }
}

for(i in 1:NROW( consumer_data.clean$Unit.Cost )){
  if( is.na(consumer_data.clean$Unit.Cost[i]) ){
     consumer_data.clean$Unit.Cost[i] <- consumer_data.clean$Total.Cost[i] / consumer_data.clean$Units.S
  }
}

for(i in 1:NROW( consumer_data.clean$Total.Revenue )){
  if( is.na(consumer_data.clean$Total.Revenue[i]) ){
     consumer_data.clean$Total.Revenue[i] <- consumer_data.clean$Unit.Price[i] * consumer_data.clean$Uni
  }
}

for(i in 1:NROW( consumer_data.clean$Total.Cost )){
  if( is.na(consumer_data.clean$Total.Cost[i]) ){
     consumer_data.clean$Total.Cost[i] <- consumer_data.clean$Unit.Cost[i] * consumer_data.clean$Units.S
  }
}

for(i in 1:NROW( consumer_data.clean$Total.Profit )){
  if( is.na(consumer_data.clean$Total.Profit[i]) ){
     consumer_data.clean$Total.Profit[i] <- consumer_data.clean$Total.Revenue[i] - consumer_data.clean$T
  }
}

# use the mean for all the values that fail to fix through calculation

consumer_data.clean[is.na(consumer_data.clean$Units.Sold),"Units.Sold"] <- mean(consumer_data.clean$Unit
consumer_data.clean[is.na(consumer_data.clean$Unit.Price),"Unit.Price"] <- mean(consumer_data.clean$Unit
consumer_data.clean[is.na(consumer_data.clean$Unit.Cost),"Unit.Cost"] <- mean(consumer_data.clean$Unit.C
consumer_data.clean[is.na(consumer_data.clean$Total.Revenue),"Total.Revenue"] <- mean(consumer_data.clea
consumer_data.clean[is.na(consumer_data.clean$Total.Cost),"Total.Cost"] <- mean(consumer_data.clean$Tota
consumer_data.clean[is.na(consumer_data.clean$Total.Profit),"Total.Profit"] <- mean(consumer_data.clean$

# replace missing dates with the most frequent dates
most_frequent_date <- mean(na.omit(consumer_data.clean$Order.Date))
consumer_data.clean[is.na(consumer_data.clean$Order.Date),"Order.Date"] <- most_frequent_date

most_frequent_date <- mean(na.omit(consumer_data.clean$Ship.Date))
consumer_data.clean[is.na(consumer_data.clean$Ship.Date),"Ship.Date"] <- most_frequent_date

apply(consumer_data.clean,2,function(c) sum(is.na(c)))
```

```
##         Region        Country              X  Sales.Channel Order.Priority
##              0              0              0              0              0
##     Order.Date       Order.ID      Ship.Date     Units.Sold     Unit.Price
##              0              0              0              0              0
##      Unit.Cost  Total.Revenue     Total.Cost   Total.Profit
##              0              0              0              0
```

d. Format/reformat the data as necessary. Please note that as you proceed through the project, you may need to do additional formatting to enable your analysis. [5]

```
#The data formating is done before
```

3. Structured Data Analysis/Modeling [35]

Write code to conduct analysis that will answer the questions below. You are encouraged to use tables/graphs where necessary to visualize results. Additionally, your code should be shown along with each question, the result and notes that explain the results.

a. What is the average spend on beverages in each country? [3]

```
data.consumer <- consumer_data.clean
data.consumer.bev <-  subset(data.consumer, data.consumer$X=="Beverages")


consumer.countries <- table(data.consumer.bev$Country)
consumer.countries <- as.data.frame(consumer.countries)

consumer.countries$avg_spend <- c(0)


for(i in 1:NROW(consumer.countries$Var1)){
  country_name = toString(consumer.countries$Var1[i])
  part_data = subset(data.consumer.bev, data.consumer.bev$Country==country_name)
  consumer.countries$avg_spend[i] <- mean(part_data$Total.Cost)
}
```

```
hist(consumer.countries$avg_spend, col="orange")
```

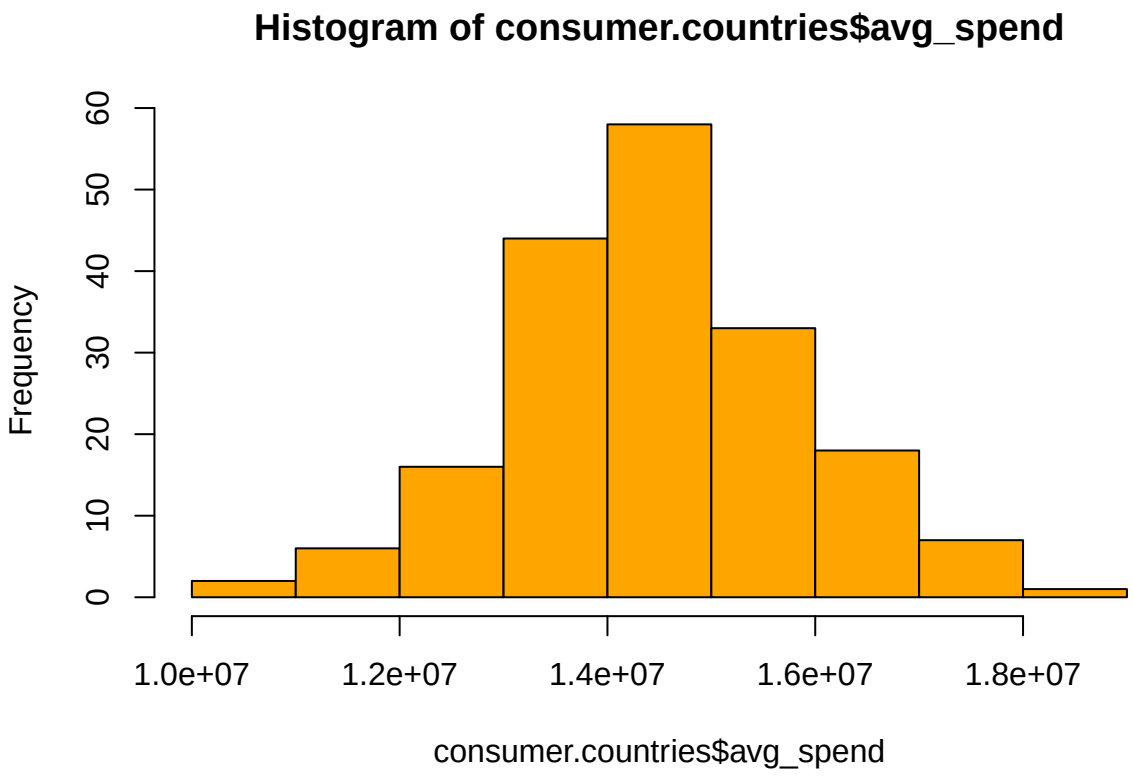


Table 1: Country and Avg Spend on beverages

Var1	Freq	avg_spend
Afghanistan	41	14214007
Albania	37	15677625
Algeria	47	11862718
Andorra	39	14645735

b. Which country has the highest spending on beverages? [2]

```
highest_spending <- data.consumer.bev$Country[match(max(data.consumer.bev$Total.Cost), data.consumer.bev
print(paste("The country with the highest spending on beverages is: ", highest_spending))
```

```
## [1] "The country with the highest spending on beverages is:  Andorra"
```

c. Which country consumes the most beverages? [2]

```
most_consume <- consumer.countries$Var1[match(max(consumer.countries$Freq), consumer.countries$Freq)]
print(paste("The country that consumes the most beverage is : ", most_consume))
```

```
## [1] "The country that consumes the most beverage is :  Finland"
```

d. What is the average profit from the sale of beverages in each country? [3]

```
consumer.countries$avg_profit <- c(0)

for(i in 1:NROW(consumer.countries$Var1)){
  country_name = toString(consumer.countries$Var1[i])
  part_data = subset(data.consumer.bev, data.consumer.bev$Country==country_name)
  consumer.countries$avg_profit[i] <- mean(part_data$Total.Profit)
}
```

```
hist(consumer.countries$avg_profit, col="orange")
```

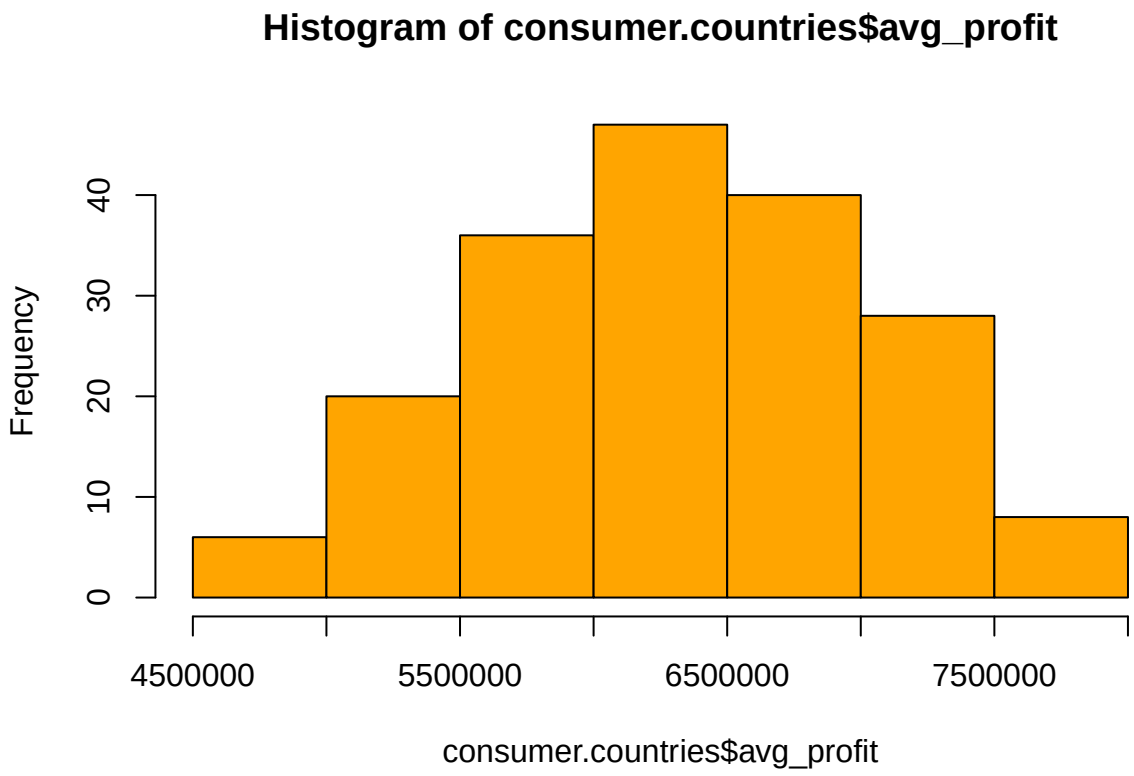


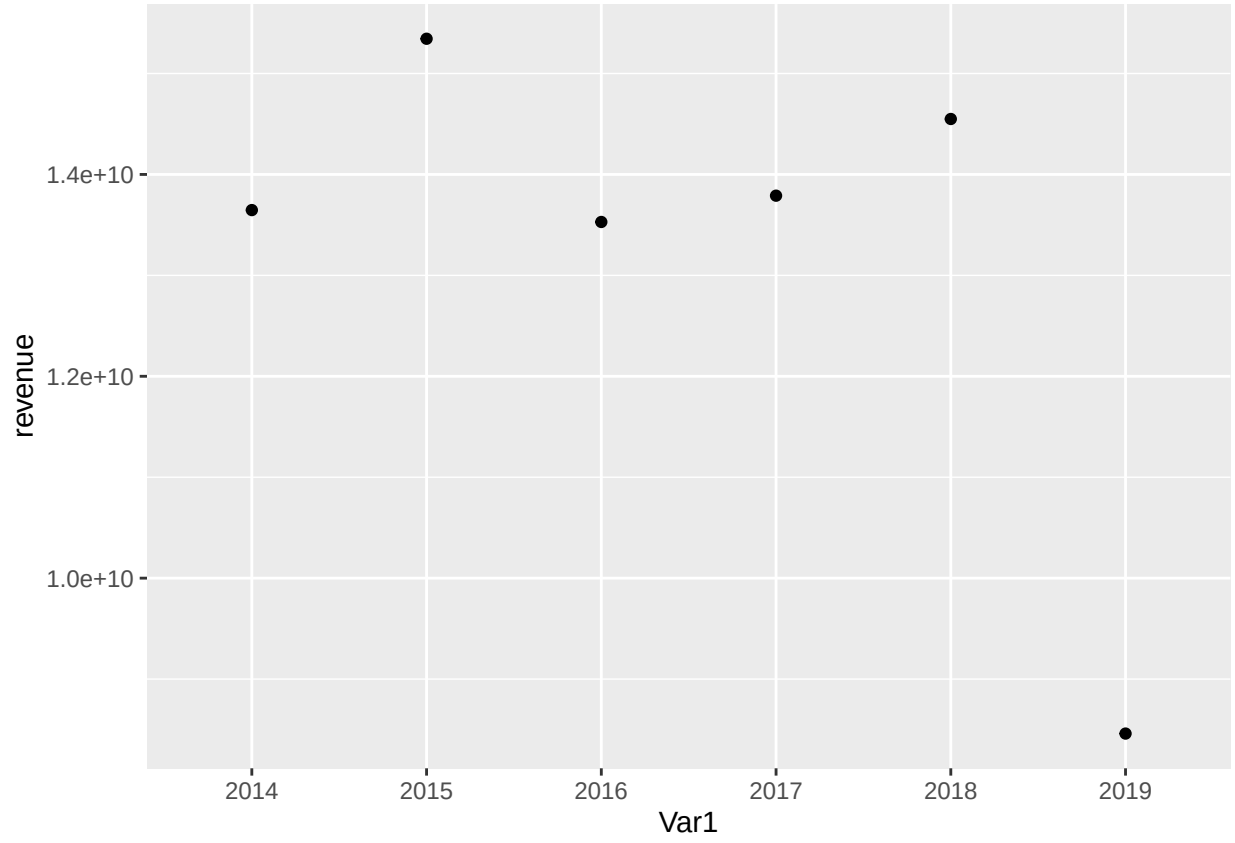
Table 2: Country and avg profit on beverages

Var1	Freq	avg_spend	avg_profit
Afghanistan	41	14214007	6591141
Albania	37	15677625	6678630
Algeria	47	11862718	5687562
Andorra	39	14645735	6821596

e. What has been the total revenue from beverages for each year since 2014? [5]

```r
# convert the order dates to years only
data.consumer.bev.yearOnly <- data.consumer.bev


data.consumer.bev.yearOnly$Order.Date <- as.integer(format(data.consumer.bev$Order.Date, format = "%Y"))

bev_yearly_rev <- table(data.consumer.bev.yearOnly$Order.Date)
bev_yearly_rev <- as.data.frame(bev_yearly_rev)

bev_yearly_rev$revenue <- c(0)

bev_yearly_rev <- subset(bev_yearly_rev, bev_yearly_rev$Var1 != 2013)
bev_yearly_rev <- subset(bev_yearly_rev, bev_yearly_rev$Var1 != 2012)

data = subset(data.consumer.bev.yearOnly, data.consumer.bev.yearOnly$Order.Date== bev_yearly_rev$Var1[1]

# calculate the revenues
for(i in 1:NROW(bev_yearly_rev$Var1)){
  data = subset(data.consumer.bev.yearOnly, data.consumer.bev.yearOnly$Order.Date== bev_yearly_rev$Var1

  bev_yearly_rev$revenue[i] <- sum(data$Total.Revenue)

}
```

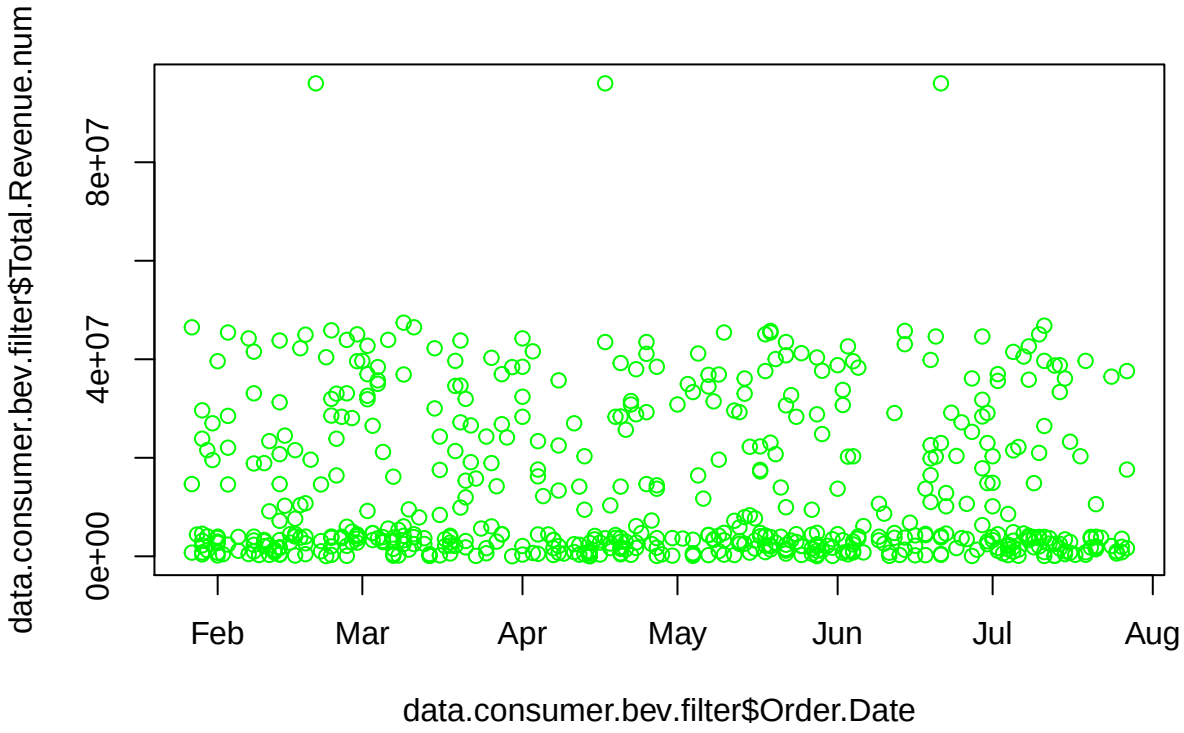
```r
bev_yearly_rev %>% ggplot(aes(x=Var1,y=revenue)) +geom_point()
```



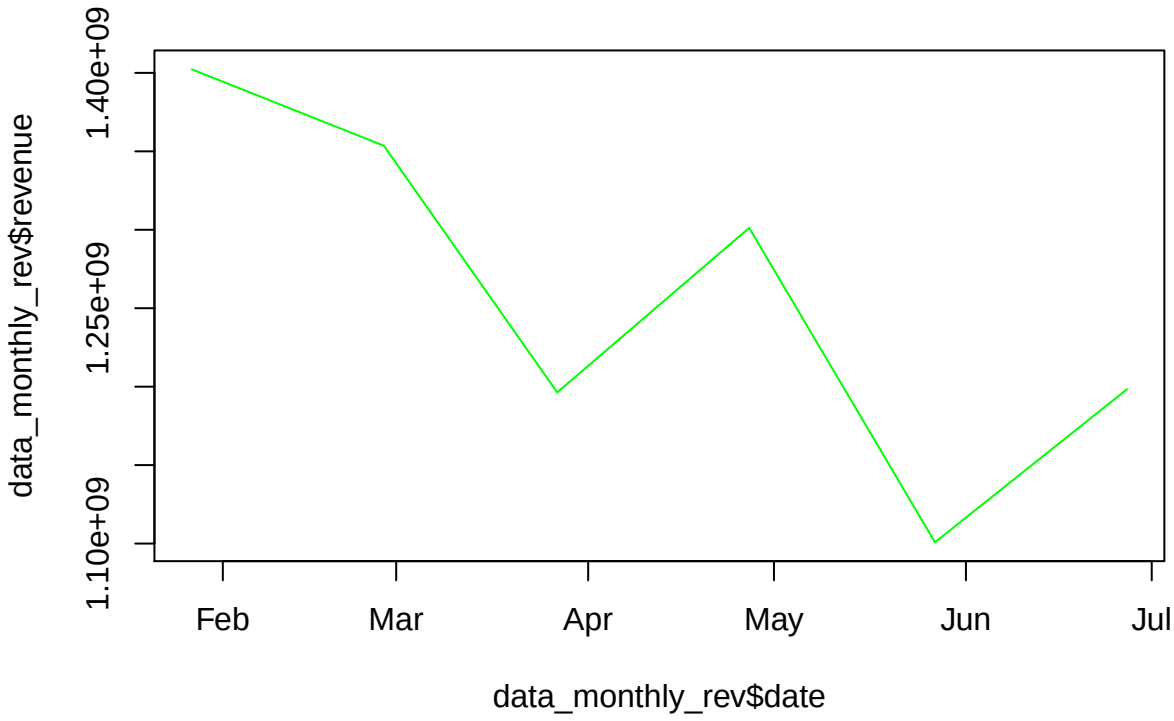
f. Plot a time series graph showing change in overall revenues from beverages for the last six months (in the dataset). [4]

```
earliest_date <- max(as.integer(format(data.consumer.bev$Order.Date, "%Y%m%d")))
earliest_date <- as.Date(toString(earliest_date), "%Y%m%d")

analysis_date <- earliest_date %m-% months(6)

data.consumer.bev.filter <- filter(data.consumer.bev,(data.consumer.bev$Order.Date >= analysis_date ))

data.consumer.bev.filter$Total.Revenue.num <- data.consumer.bev.filter$Total.Revenue

data_monthly_rev <- data.frame(date = c(as.Date("2014-12-31")), revenue = c(0,0,0,0,0,0))


# avg each month data
for(i in 1:6){
  start_mth <- analysis_date %m+% months(i-1)
  end_mth <- analysis_date %m+% months(i)
  fil_data <- filter(data.consumer.bev,(data.consumer.bev$Order.Date >= start_mth & data.consumer.bev$O
  avg_rev <- sum(fil_data$Total.Revenue)
  data_monthly_rev$date[i] <- start_mth
  data_monthly_rev$revenue[i] <- avg_rev
}
```

```
# The revenues for each day
plot(data.consumer.bev.filter$Total.Revenue.num~data.consumer.bev.filter$Order.Date,type="p", col="gree
```




```
# Total revenue per month
plot(data_monthly_rev$revenue~data_monthly_rev$date,type="l", col="green")
```



g. What is the dominant sales channel for beverages?[2]

```
sales_channels <- table(data.consumer.bev$Sales.Channel)
sales_channels <- as.data.frame(sales_channels)
dominantChannel <- sales_channels$Var1[match(max(sales_channels$Freq), sales_channels$Freq)]
```

```
## [1] "The most dominant sales channel is :  Online"
```

h. Determine whether beverages units sold is above the overall average for units sold for all other products. [3]

```
data.consumer.not.bev <- subset(data.consumer, data.consumer$X!="Beverages")
beverages_unit_sold <- mean(data.consumer.bev$Units.Sold)
overall_unit_sold_except_bev <- mean(data.consumer.not.bev$Units.Sold)
```

```
## [1] "Beverage units sold is greater than the other products with beverages at:  5032.22624434389  ,ar
```

i. In which season (Spring, Summer, Autumn, Winter) does persons spend the most on beverages? [6]

```
data.consumer.bev.m_n_d <- data.consumer.bev
data.consumer.bev.m_n_d$m_n_d <- format(data.consumer.bev$Order.Date, "%m%d")

spring_data <- filter(data.consumer.bev,  data.consumer.bev.m_n_d$m_n_d>= "0301" & data.consumer.bev.m_
```

```
summer_data <- filter(data.consumer.bev,  data.consumer.bev.m_n_d$m_n_d>= "0601" & data.consumer.bev.m_

autumn_data <- filter(data.consumer.bev,  data.consumer.bev.m_n_d$m_n_d>= "0901" & data.consumer.bev.m_

winter_data <- filter(data.consumer.bev,  (data.consumer.bev.m_n_d$m_n_d>= "1201" & data.consumer.bev.m

spring.spend <- sum(as.numeric(spring_data$Total.Cost))
summer.spend <- sum(as.numeric(summer_data$Total.Cost))
autumn.spend <- sum(as.numeric(autumn_data$Total.Cost))
winter.spend <- sum(as.numeric(winter_data$Total.Cost))

seasons_spend <- data.frame(spring.spend, summer.spend, autumn.spend, winter.spend)

most_prod_mth <- names(seasons_spend[match(max(seasons_spend),seasons_spend)])
```

```
## [1] "The season when people spend the most on beverages is :  spring.spend"
```

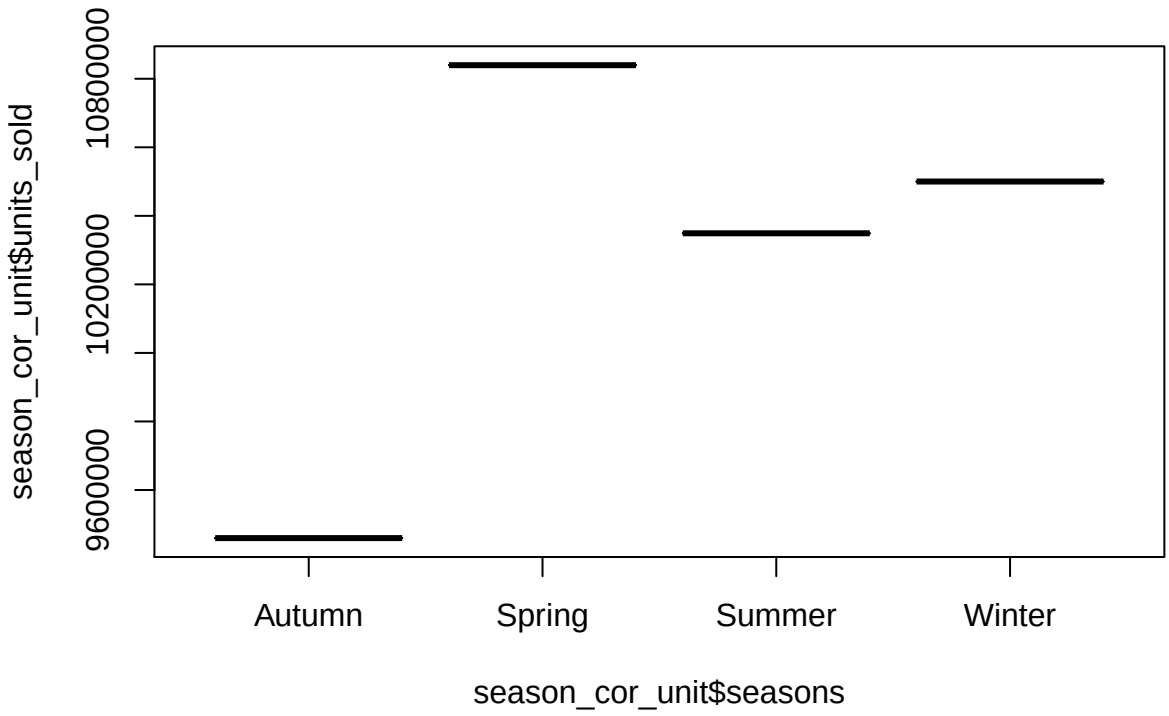
j. Is there a correlation between the season and the units sold for beverages? Explain the result. [5]

The graph and the correlation details below is able to show us how the two variables poorly correlates. The seasons drastically causes the number of units sold to change creating large gaps between each season. The cor test show us that they correlates by only 0.8% which is very low and very high chance of recieveing an incorrect prediction of 45.3%. Therefore, from the analysis it can be said that the two variable does not correlates sttistically.

```
seasons <- c("Spring", "Summer", "Autumn", "Winter")

season_cor_unit <- data.frame(seasons)
season_cor_unit$units_sold <- c(0)

season_cor_unit$units_sold[1] <- sum(spring_data$Units.Sold)
season_cor_unit$units_sold[2] <- sum(summer_data$Units.Sold)
season_cor_unit$units_sold[3] <- sum(autumn_data$Units.Sold)
season_cor_unit$units_sold[4] <- sum(winter_data$Units.Sold)

# add season to each row
data.consumer.bev.m_n_d$season <- c(0)

data.consumer.bev.m_n_d$m_n_d <- na.omit(data.consumer.bev.m_n_d$m_n_d)
for(i in 1:NROW(data.consumer.bev.m_n_d$season)){

  if(data.consumer.bev.m_n_d$m_n_d[i]>= "0301" & data.consumer.bev.m_n_d$m_n_d[i]<="0531"){
    data.consumer.bev.m_n_d$season[i] <- as.numeric(1)

  }else if(data.consumer.bev.m_n_d$m_n_d[i]>= "0601" & data.consumer.bev.m_n_d$m_n_d[i]<="0831"){
    data.consumer.bev.m_n_d$season[i] <-  as.numeric(2)

  }else if(data.consumer.bev.m_n_d$m_n_d[i]>= "0901" & data.consumer.bev.m_n_d$m_n_d[i]<="1130"){
    data.consumer.bev.m_n_d$season[i] <-  as.numeric(3)

  }else{
    data.consumer.bev.m_n_d$season[i] <-  as.numeric(4)
```

```
  }
}
```

```
plot(season_cor_unit$units_sold~season_cor_unit$seasons,type="l", col="green")
```




```
#The correlation test
# 1 represents Spring , 2 Summer, 3 Auntumn, 4 Winter
cor.test(data.consumer.bev.m_n_d$Units.Sold, data.consumer.bev.m_n_d$season)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data.consumer.bev.m_n_d$Units.Sold and data.consumer.bev.m_n_d$season
## t = 0.75047, df = 8175, p-value = 0.453
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.01337763  0.02996975
## sample estimates:
##         cor
## 0.008299956
```

4. Recommendation:

a. Based on your analysis of both the tweet data and structured data, what would you recommend to Hard Knocks and why?

I would recommend that hard knocks do open their next franchise because they have have positive sentiments based on the both datasets used. In order for Hard Knocks to maximize profits, based on the tweet data, focus on Atlanta GA, United States, London England and Chicago IL because they were the locations with the most positive tweets fo Beer, Beverage, Party and Concert respectively. Additionally, they should seek to understand why New York, Ny has a dominant emotion of angry and come up with a solution as to how to deal with it. In relation to the structured data, focus on Afghanistan, Albina, Algeria,Andorra because they spend the highest on beverages.

5. BONUS – 10 marks

a. Which features in the dataset can be used to predict the units sold for beverages?

The results below shows that neither the Region nor the order priority can be used to predict the number of units sold. However, the best of the two predictors is the Region that the order was made. This predictor shows 28.86% failure rate. This value is very high and is far above the optimal case. In addition, it correlates with the number of units sold by 1%. It is a poor predictor but the best of the two.

```
OrderPriorities <- count(data.consumer.bev.m_n_d,Order.Priority)

data.consumer.bev.m_n_d$Order.Priority.num <- c(0)


for(i in 1:NROW(data.consumer.bev.m_n_d$Order.Priority)){
  data.consumer.bev.m_n_d$Order.Priority.num[i] <- as.numeric(match(data.consumer.bev.m_n_d$Order.Prior
}

country_count <- count(data.consumer.bev.m_n_d, Region)

data.consumer.bev.m_n_d$Country.num <- c(0)


for(i in 1:NROW(data.consumer.bev.m_n_d$Region)){
  data.consumer.bev.m_n_d$Country.num[i] <- as.numeric(match(data.consumer.bev.m_n_d$Region[i], country
}

cor.test(data.consumer.bev.m_n_d$Units.Sold, data.consumer.bev.m_n_d$Country.num)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data.consumer.bev.m_n_d$Units.Sold and data.consumer.bev.m_n_d$Country.num
## t = 1.0612, df = 8175, p-value = 0.2886
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.009941261  0.033403134
## sample estimates:
##        cor
## 0.01173645
```

```
cor.test(data.consumer.bev.m_n_d$Units.Sold, data.consumer.bev.m_n_d$Order.Priority.num)
```

```
##
```

```
##  Pearson's product-moment correlation
##
## data:  data.consumer.bev.m_n_d$Units.Sold and data.consumer.bev.m_n_d$Order.Priority.num
## t = -0.14704, df = 8175, p-value = 0.8831
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.02330066  0.02004959
## sample estimates:
##          cor
## -0.001626302
```