install.packages("data.table")

#### Load required libraries

library(data.table)

library(ggplot2)

library(ggmosaic)

library(readr)

library(readxl)

####loading and assign the data files to data.tables

filePath <- "intership/"

transactionData <- read\_excel("intership/QVI\_transaction\_data.xlsx")

transactionData <- data.table(transactionData)

customerData <- fread(paste0(filePath,"QVI\_purchase\_behaviour.csv"))

#### Examine transaction data

str(transactionData)

head(transactionData)

view(transactionData)

#### Examine customer data

str(customerData)

head(customerData)

view(customerData)

# finding missing values using short sum()function.

sum(is.na(transactionData)) # result: 0

sum(is.na(customerData)) # result: 0

#### Convert DATE column to a date format

transactionData$DATE <- as.Date(transactionData$DATE, origin = "1899-12-30")

#### Examining PROD\_NAME

summary(transactionData$PROD\_NAME)

head(transactionData$PROD\_NAME)

transactionData[, .N, PROD\_NAME]

productWords <- data.table(unlist(strsplit(unique(transactionData[, PROD\_NAME]), " ")))

setnames(productWords, 'words')

###let's remove all words with digits and special characters such as '&' from our set of product words

productWords <- productWords[grepl("\\d", words) == FALSE, ]

# Remove special characters

productWords <- productWords[grepl("[:alpha:]", words), ]

###counting the number of times a word appears and sorting first 10 highest.

head(productWords[, .N, words][order(N, decreasing = TRUE)], 10)

# remove the salsa product

transactionData[, SALSA := grepl("salsa", tolower(PROD\_NAME))]

transactionData <- transactionData[SALSA == FALSE, ][, SALSA := NULL]

# summarizing the data

summary(transactionData)

# Filter the dataset to find the out-liers.

transactionData[PROD\_QTY == 200, ]

#Let's see if the customer has had other transactions.

transactionData[LYLTY\_CARD\_NBR == 226000, ]

# Filter out the customer based on the loyalty card number

transactionData <- transactionData[LYLTY\_CARD\_NBR != 226000, ]

#### Re‐examine transaction data

summary(transactionData)

#### Count the number of transactions by date

transactionData[, .N, by = DATE] #[order(DATE)], sort if you need.

#### Create a sequence of dates and join this the count of transactions by date

allDates <- data.table(seq(as.Date("2018/07/01"), as.Date("2019/06/30"), by = "day"))

setnames(allDates, "DATE")

transactions\_by\_day <- merge(allDates, transactionData[, .N, by = DATE], all.x = TRUE)

### Setting plot themes to format graphs

theme\_set(theme\_bw())

theme\_update(plot.title = element\_text(hjust = 0.5))

# Plot transactions over time

ggplot(transactions\_by\_day, aes(x = DATE, y = N)) +

geom\_line(col = "orange") +

labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +

scale\_x\_date(breaks = "1 month") +

theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))

#### Filter to December and look at individual days

ggplot(transactions\_by\_day[month(DATE) == 12, ], aes(x = DATE, y = N)) +

geom\_line() +

labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +

scale\_x\_date(breaks = "1 day") +

theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))

#### We can work this out by taking the digits that are in PROD\_NAME

transactionData[, PACK\_SIZE := parse\_number(PROD\_NAME)]

# Let's check if the pack sizes look sensible

transactionData[, .N, PACK\_SIZE][order(PACK\_SIZE, decreasing = TRUE)]

#Plot a histogram showing the number of transactions by pack size.

options(scipen=999) # turn off scientific notations like 1e+05

hist(transactionData[, PACK\_SIZE], col = "green",border = "red" , xlab = "PACK SIZE", ylab = "Total no of chips purchased", main = "HISTOGRAM OF NO. OF CHIPS PURCHASED ACCORDING TO THEIR PACK SIZES")

#### Brands

transactionData[, BRAND := toupper(substr(PROD\_NAME, 1, regexpr(pattern = ' ', PROD\_NAME) - 1))]

# Checking brands

transactionData[, .N, by = BRAND][order(-N)]

#### Clean brand names

transactionData[BRAND == "RED", BRAND := "RRD"]

#other similar brands

transactionData[BRAND == "SNBTS", BRAND := "SUNBITES"]

transactionData[BRAND == "INFZNS", BRAND := "INFUZIONS"]

transactionData[BRAND == "WW", BRAND := "WOOLWORTHS"]

transactionData[BRAND == "SMITH", BRAND := "SMITHS"]

transactionData[BRAND == "NCC", BRAND := "NATURAL"]

transactionData[BRAND == "DORITO", BRAND := "DORITOS"]

transactionData[BRAND == "GRAIN", BRAND := "GRNWVES"]

# Checking

transactionData[, .N, by = BRAND][order(BRAND)]

#### Examining customer data

str(customerData)

head(customerData)

## Examining key values

customerData[, .N, by = LIFESTAGE][order(-N)]

customerData[, .N, by = PREMIUM\_CUSTOMER][order(-N)]

#### Merge transaction data to customer data

data <- merge(transactionData, customerData, all.x = TRUE)

#Check for missing customer details

sum(is.null(data)) #use also; colSums(is.na(data)), it's perfect than that one.

# Save dataset as a csv.

write.csv(data,file="intership/QVI\_data.csv")

#### Total sales by LIFESTAGE and PREMIUM\_CUSTOMER

sales <- data[, .(SALES = sum(TOT\_SALES)), .(LIFESTAGE, PREMIUM\_CUSTOMER)]

# create plot

p <- ggplot(data = sales) +

geom\_mosaic(aes(weight = SALES, x = product(PREMIUM\_CUSTOMER, LIFESTAGE) , fill = PREMIUM\_CUSTOMER)) +

labs(x = "Lifestage", y = "Premium customer flag", title = "Proportion of sales") + theme(axis.text.x = element\_text(angle = 50, vjust = 0.5, size = 10))

# Plot and label with proportion of sales

p +

geom\_text(data = ggplot\_build(p)$data[[1]], aes(x = (xmin + xmax)/2 , y = (ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)\*100, '%'))))

#### Number of customers by LIFESTAGE and PREMIUM\_CUSTOMER

customers <- data[, .(CUSTOMERS = uniqueN(LYLTY\_CARD\_NBR)), .(LIFESTAGE, PREMIUM\_CUSTOMER)][order(-CUSTOMERS)]

labels <- c("A", "b", "c", "D", "e", "f", "g")

# Create plot

p <- ggplot(data = customers) + geom\_mosaic(aes(weight = CUSTOMERS, x = product(PREMIUM\_CUSTOMER, LIFESTAGE), fill = PREMIUM\_CUSTOMER)) + labs(x = "Lifestage", y = "Premium customer flag", title = "Proportion of customers") + theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))+scale\_x\_productlist(labels = labels )

p + geom\_text(data = ggplot\_build(p)$data[[1]], aes(x = (xmin + xmax)/2 , y = (ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)\*100,'%'))))

#### Average number of units per customer by LIFESTAGE and PREMIUM\_CUSTOMER

avg\_units <- data[, .(AVG = sum(PROD\_QTY)/uniqueN(LYLTY\_CARD\_NBR)), .(LIFESTAGE, PREMIUM\_CUSTOMER)][order(-AVG)]

ggplot(data = avg\_units, aes(weight = AVG, x = LIFESTAGE, fill = PREMIUM\_CUSTOMER)) + geom\_bar(position = position\_dodge()) +

labs(x = "Lifestage", y = "Avg units per transaction", title = "Units per customer") + theme(axis.text.x = element\_text(angle = 90, vjust = 0.75, size = 7))

#### Average price per unit by LIFESTAGE and PREMIUM\_CUSTOMER

avg\_price <- data[, .(AVG = sum(TOT\_SALES)/sum(PROD\_QTY)), .(LIFESTAGE, PREMIUM\_CUSTOMER)][order(-AVG)]

#### Create plot

ggplot(data = avg\_price, aes(weight = AVG, x = LIFESTAGE, fill = PREMIUM\_CUSTOMER)) + geom\_bar(position = position\_dodge()) + labs(x = "Lifestage", y = "Avg price per unit", title = "Price per unit") + theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))

#### young singles and couples

pricePerUnit <- data[, price := TOT\_SALES/PROD\_QTY]

t.test(data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER == "Mainstream", price]

, data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER != "Mainstream", price]

, alternative = "greater")

**Answer for the t-test**

Welch Two Sample t-test

data: data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER == "Mainstream", price] and data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER != "Mainstream", price]

t = 37.624, df = 54791, p-value < 2.2e-16

alternative hypothesis: true difference in means is greater than 0

95 percent confidence interval:

0.3187234 Inf

sample estimates:

mean of x mean of y

4.039786 3.706491

### Deeping dive into specific customer segments for insights

segment1 <- data[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM\_CUSTOMER == "Mainstream",]

other <- data[!(LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM\_CUSTOMER == "Mainstream"),]

### Brand

quantity\_segment1 <- segment1[, sum(PROD\_QTY)]

quantity\_other <- other[, sum(PROD\_QTY)]

quantity\_segment1\_by\_brand <- segment1[, .(targetSegment = sum(PROD\_QTY)/quantity\_segment1), by = BRAND]

quantity\_other\_by\_brand <- other[, .(other = sum(PROD\_QTY)/quantity\_other), by = BRAND]

brand\_proportions <- merge(quantity\_segment1\_by\_brand, quantity\_other\_by\_brand)[, affinityToBrand := targetSegment/other]

brand\_proportions[order(-affinityToBrand)]

#### Deeping dive into Mainstream, young singles/couples

quantity\_segment1\_by\_pack <- segment1[, .(targetSegment = sum(PROD\_QTY)/quantity\_segment1), by = PACK\_SIZE]

quantity\_other\_by\_pack <- other[, .(other = sum(PROD\_QTY)/quantity\_other), by = PACK\_SIZE]

pack\_proportions <- merge(quantity\_segment1\_by\_pack, quantity\_other\_by\_pack)[, affinityToPack := targetSegment/other]

pack\_proportions[order(-affinityToPack)]

#### Preferred pack size compared to the rest of the population

quantity\_segment1\_by\_pack <- segment1[, .(targetSegment = sum(PROD\_QTY)/quantity\_segment1), by = PACK\_SIZE]

quantity\_other\_by\_pack <- other[, .(other = sum(PROD\_QTY)/quantity\_other), by = PACK\_SIZE]

pack\_proportions <- merge(quantity\_segment1\_by\_pack, quantity\_other\_by\_pack)[, affinityToPack := targetSegment/other]

pack\_proportions[order(-affinityToPack)]