```
install.packages("data.table")
install.packages("tidyverse")

library(data.table)
library(ggplot2)
library(ggmosaic)
library(readr)
library(readxl)
library(tidyverse)
library(dplyr)

filePath <- "C:/Users/olade/OneDrive/Documents/Data-Analytics/Internship/"
### transactionData <- fread(paste0(filePath,"QVI_transaction_data.xlsx"))
### customerData <- fread(paste0(filePath,"QVI_purchase_behaviour.csv"))

customerData <- read_csv("Data-Analytics/Internship/QVI_purchase_behaviour.csv")
transactionData <- read_excel("Data-Analytics/Internship/QVI_transaction_data.xlsx")

transactionData <- as.data.table(transactionData)
customerData <- as.data.table(customerData)

View(transactionData)
View(customerData)

str(customerData)
str(transactionData)

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

transactionData

transactionData$PROD_NAME

table(transactionData$PROD_NAME)

unique(transactionData$PROD_NAME)

summary(transactionData$PROD_NAME)

transactionData %>% count(PROD_NAME, sort = TRUE)

transactionData %>%
  count(PROD_NAME, sort = TRUE) %>%
  top_n(10, n) %>%
  ggplot(aes(x = reorder(PROD_NAME, n), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue") +
  coord_flip() +
  labs(title = "Top Product Names", x = "", y = "") +
  theme_minimal()


productWords <- data.table(unlist(strsplit(unique(transactionData$PROD_NAME), " ")))
setnames(productWords, 'words')
## View(productWords)

cleanWords <- productWords[!grepl("[^a-zA-Z]", words)]
## View(cleanWords)

wordFreq <- cleanWords[, .N, by = words][order(-N)]
## View(wordFreq)

transactionData[, SALSA := grepl("salsa", tolower(PROD_NAME))]
transactionData <- transactionData[SALSA == FALSE, ][, SALSA := NULL]
```

```
summary(transactionData)

transactionData[order(-PROD_QTY)][1:5]
transactionData[PROD_QTY == 200]

outlier_cust <- transactionData[PROD_QTY == 200, unique(LYLTY_CARD_NBR)]

transactionData[LYLTY_CARD_NBR == outlier_cust]

transactionData_clean <- transactionData[LYLTY_CARD_NBR != outlier_cust]


summary(transactionData_clean)
transactions_by_day <- transactionData_clean[, .N, by = DATE][order(DATE)]

all_dates <- data.table(DATE = seq(as.Date("2018-07-01"), as.Date("2019-06-30"), by = "day"))
transactions_full <- merge(all_dates, transactions_by_day, by = "DATE", all.x = TRUE)

theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))

ggplot(transactions_full, aes(x = DATE, y = N)) +
  geom_line() +
  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))


december_data <- transactions_full[format(transactions_full$DATE, "%m") == "12", ]

ggplot(december_data, aes(x = DATE, y = N)) +
  geom_line() +
  labs(x = "Day", y = "Number of Transactions", title = "Transactions in December") +
  scale_x_date(breaks = "1 day", date_labels = "%d") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))

transactionData[, PACK_SIZE := parse_number(PROD_NAME)]
transactionData[, .N, PACK_SIZE][order(PACK_SIZE)]


ggplot(transactionData, aes(x = PACK_SIZE)) +
  geom_histogram(binwidth = 10, fill = "skyblue", color = "black", alpha = 0.7) +
  labs(x = "Pack Size (grams)", y = "Number of Transactions", title = "Transactions by Pack Size")
+
  theme_minimal()



transactionData[, BRAND := tstrsplit(PROD_NAME, " ")[[1]]]

transactionData[, .N, by = BRAND][order(-N)]
#### Clean brand names
transactionData[BRAND %in% c("RED", "Red"), BRAND := "RRD"]
transactionData[BRAND %in% c("INFZNS", "Infzns"), BRAND := "Infuzions"]
transactionData[BRAND %in% c("SMITH", "Smith"), BRAND := "Smiths"]
transactionData[BRAND %in% c("DORITO", "Dorito"), BRAND := "Doritos"]
transactionData[BRAND %in% c("GRNWAVS", "GrnWves", "Grain"), BRAND := "GrnWaves"]
transactionData[BRAND %in% c("NCC", "NATURAL", "Natural"), BRAND := "Natural"]

# View the cleaned list
transactionData[, .N, by = BRAND][order(-N)]

#### Check the cleaned brand distribution
transactionData[, .N, by = BRAND][order(-N)]
```

```
summary(customerData)
str(customerData)
colSums(is.na(customerData))
customerData[, .N, by = LIFESTAGE][order(-N)]
customerData[, .N, by = PREMIUM_CUSTOMER][order(-N)]
hist(customerData$LYLTY_CARD_NBR, main = "Distribution of Age", xlab = "lylty_card_nbr", col =
"skyblue")


ggplot(customerData, aes(x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) +
  geom_bar(position = "dodge") +
  theme_minimal() +
  labs(title = "Customer Distribution by Lifestage and Premium Status")


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

colSums(is.na(data))
fwrite(data, paste0(filePath,"QVI_data.csv"))


transactionData <- merge(transactionData, customerData,
                         by = "LYLTY_CARD_NBR",
                         all.x = TRUE)

sales_by_segment <- transactionData[, .(TOTAL_SALES = sum(TOT_SALES)),
                                    by = .(LIFESTAGE, PREMIUM_CUSTOMER)][order(-TOTAL_SALES)]

ggplot(sales_by_segment, aes(x = LIFESTAGE, y = TOTAL_SALES, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Total Chip Sales by Customer Segment",
       x = "Lifestage",
       y = "Total Sales ($)",
       fill = "Premium Customer") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Total quantity purchased per customer
customer_qty <- transactionData[, .(TOTAL_QTY = sum(PROD_QTY)), by = .(LYLTY_CARD_NBR, LIFESTAGE,
PREMIUM_CUSTOMER)]

# Average quantity per customer by segment
avg_qty_by_segment <- customer_qty[, .(AVG_QTY = mean(TOTAL_QTY)), by = .(LIFESTAGE,
PREMIUM_CUSTOMER)]


ggplot(avg_qty_by_segment, aes(x = LIFESTAGE, y = AVG_QTY, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Chips Purchased per Customer by Segment",
       x = "Lifestage",
       y = "Average Quantity",
       fill = "Premium Customer") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))


# Calculate price per unit for each transaction
transactionData[, UNIT_PRICE := TOT_SALES / PROD_QTY]

# Now average the unit price by LIFESTAGE and PREMIUM_CUSTOMER
avg_price_segment <- transactionData[, .(AVG_UNIT_PRICE = mean(UNIT_PRICE)),
                                     by = .(LIFESTAGE, PREMIUM_CUSTOMER)]


ggplot(avg_price_segment, aes(x = LIFESTAGE, y = AVG_UNIT_PRICE, fill = PREMIUM_CUSTOMER)) +
```

```
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Unit Price of Chips by Customer Segment",
       x = "Lifestage",
       y = "Average Price ($)",
       fill = "Premium Customer") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))


# Filter to only include relevant LIFESTAGEs and PREMIUM_CUSTOMER types
filtered_data <- transactionData[
  LIFESTAGE %in% c("MIDAGE SINGLES/COUPLES", "YOUNG SINGLES/COUPLES") &
    PREMIUM_CUSTOMER %in% c("Mainstream", "Premium")
]


filtered_data[, UNIT_PRICE := TOT_SALES / PROD_QTY]

# Perform t-test
t_test_result <- t.test(UNIT_PRICE ~ PREMIUM_CUSTOMER, data = filtered_data)
print(t_test_result)

ggplot(filtered_data, aes(x = PREMIUM_CUSTOMER, y = UNIT_PRICE)) +
  geom_boxplot() +
  facet_wrap(~LIFESTAGE) +
  labs(title = "Unit Price Comparison: Premium vs Mainstream",
       x = "Customer Segment", y = "Unit Price ($)") +
  theme_minimal()



# Step 1: Filter for the segment
mainstream_young <- transactionData[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER ==
"Mainstream"]

# Step 2: Count brand preference in the segment
segment_brand_counts <- mainstream_young[, .N, by = BRAND]
segment_brand_counts[, SegmentPerc := N / sum(N)]

# Step 3: Count overall brand preference
overall_brand_counts <- transactionData[, .N, by = BRAND]
overall_brand_counts[, OverallPerc := N / sum(N)]

# Step 4: Merge and calculate affinity
affinity <- merge(segment_brand_counts, overall_brand_counts, by = "BRAND")
affinity[, AffinityScore := SegmentPerc / OverallPerc]

# Step 5: Sort by affinity score
affinity[order(-AffinityScore)]


top_affinity <- affinity[order(-AffinityScore)][1:10]  # Top 10

ggplot(top_affinity, aes(x = reorder(BRAND, AffinityScore), y = AffinityScore)) +
  geom_col(fill = "#20B2AA") +
  coord_flip() +
  labs(title = "Top Brands Preferred by Mainstream Young Singles/Couples",
       x = "Brand",
       y = "Affinity Score") +
  theme_minimal()


# Step 1: Filter for the target segment
mainstream_young <- transactionData[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER ==
"Mainstream"]
```

```
# Step 2: Count pack sizes in the segment
segment_pack_counts <- mainstream_young[, .N, by = PACK_SIZE]
segment_pack_counts[, SegmentPerc := N / sum(N)]

# Step 3: Count pack sizes in the entire population
overall_pack_counts <- transactionData[, .N, by = PACK_SIZE]
overall_pack_counts[, OverallPerc := N / sum(N)]

# Step 4: Merge and calculate affinity
pack_affinity <- merge(segment_pack_counts, overall_pack_counts, by = "PACK_SIZE")
pack_affinity[, AffinityScore := SegmentPerc / OverallPerc]

# Step 5: Sort and inspect
pack_affinity[order(-AffinityScore)]


top_pack_affinity <- pack_affinity[order(-AffinityScore)][1:10]  # Top 10

ggplot(top_pack_affinity, aes(x = reorder(as.factor(PACK_SIZE), AffinityScore), y =
AffinityScore)) +
  geom_col(fill = "#FFA500") +
  coord_flip() +
  labs(title = "Top Pack Sizes Preferred by Mainstream Young Singles/Couples",
       x = "Pack Size (g)",
       y = "Affinity Score") +
  theme_minimal()
```