Quantium Chips Customer Analysis

Swaraj Borhade

2025-06-18

— 1. Setup and Package Loading —

Run these installations once if you haven't already

install.packages("tidyverse")

install.packages("skimr")

install.packages("tinytex")

tinytex::install_tinytex() # Run this after installing tinytex package

Load necessary libraries for your analysis

library(tidyverse) library(skimr) # For enhanced data summaries

— 2. Data Loading —

Make sure these CSV files are in the same directory as your R Markdown file

transaction_data <- read_csv("QVI_transaction_data.xlsx - in.csv") purchase_behaviour <- read_csv("QVI_purchase_bel

Optional: View first few rows of each dataset to confirm loading

head(transaction_data)

head(purchase_behaviour)

— 3. Initial Data Checks —

High-Level Summaries

glimpse(transaction_data) glimpse(purchase_behaviour)

summary(transaction_data) summary(purchase_behaviour)

skim(transaction_data) # Comprehensive summary skim(purchase_behaviour)

Check and Correct Data Formats

Example: Convert DATE column to actual Date type if it's not already

Adjust the format string if your date format is different (e.g., "%d/%m/%Y")

transaction_data <- transaction_data %>% mutate(DATE = as.Date(DATE, format = "%Y-%m-%d"))

Check unique values for categorical columns to identify inconsistencies

transaction_data %>% distinct(PROD_NAME) %>% print(n = Inf) # Print all unique product names purchase_behaviour %>% distinct(LIFESTAGE) purchase_behaviour %>% distinct(PREMIUM_CUSTOMER)

— 4. Finding Outliers and Cleaning Data —

Example: Identify potential outliers in PROD_QTY (e.g., quantity = 200)

A quantity of 200 is highly unusual for a single chip purchase and likely an error.

transaction_data %>% ggplot(aes(x = PROD_QTY)) + geom_histogram(binwidth = 1, fill = "green", color = "black") + ggtitle("Histogram of Product Quantity")

Filter out the transaction with PROD_QTY of 200 (or any other identified outlier)

You might need to adjust this filter based on your data exploration

transaction_data_cleaned <- transaction_data %>% filter(PROD_QTY < 100) # Assuming quantities over 100 are outliers for chips

Verify filtering

transaction_data_cleaned %>%

ggplot(aes(x = PROD_QTY)) +

geom_histogram(binwidth = 1, fill = "green", color = "black") +

ggtitle("Histogram of Product Quantity (Cleaned)")

— 5. Feature Engineering —

Derive PACK_SIZE from PROD_NAME

Extracts the numeric value followed by 'g'

transaction_data_processed <- transaction_data_cleaned %>% mutate(PACK_SIZE = as.numeric(str_extract(PROD_NA "\d+g")))

Verify the new PACK_SIZE column

transaction_data_processed %>%

distinct(PACK_SIZE) %>%

arrange(PACK_SIZE)

Derive BRAND from PROD_NAME

This is a basic extraction; you might need to refine this with more complex logic

(e.g., specific rules for known brands if the initial word isn't always the brand)

transaction_data_processed <- transaction_data_processed %>% mutate(BRAND = str_extract(PROD_NAME, "[1]+"))

Verify the new BRAND column

transaction_data_processed %>%

distinct(BRAND) %>%

print(n = Inf) # Print all unique brands

Combine transaction data with customer behaviour data

Assuming L_CUSTOMER_ID is the common column

merged_data <- inner_join(transaction_data_processed, purchase_behaviour, by = "L_CUSTOMER_ID")

[1] A-Za-z

View structure of merged data

glimpse(merged_data)

— 6. Define Metrics of Interest and Perform Analysis by Segment —

Calculate key metrics grouped by LIFESTAGE and PREMIUM_CUSTOMER

customer_segment_summary <- merged_data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>% summarise(Total_Spend = sum(TOT_SALES), Avg_Spend_Per_Customer = sum(TOT_SALES) / n_distinct(L_CUSTOMER_ID), # Average spend by unique customer Total_Transactions = n(), Avg_Units_Per_Transaction = mean(PROD_QTY), Avg_Price_Per_Unit = mean(TOT_SALES / PROD_QTY, na.rm = TRUE), Unique_Customers = n_distinct(L_CUSTOMER_ID), .groups = 'drop' # Important to ungroup after summarise)

print(customer_segment_summary)

Analyze preferred Pack Sizes by Segment

pack_size_by_segment <- merged_data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER, PACK_SIZE) %>% summarise(Sales = sum(TOT_SALES), .groups = 'drop_last') %>% # group_by for slice_head arrange(LIFESTAGE, PREMIUM_CUSTOMER, desc(Sales)) %>% slice_head(n = 3) # Get top 3 pack sizes per segment

print(pack_size_by_segment)

Analyze preferred Brands by Segment

brand_by_segment <- merged_data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER, BRAND) %>% summarise(Sales = sum(TOT_SALES), .groups = 'drop_last') %>% # group_by for slice_head arrange(LIFESTAGE, PREMIUM_CUSTOMER, desc(Sales)) %>% slice_head(n = 3) # Get top 3 brands per segment

print(brand_by_segment)

— 7. Visualization Examples —

Use ggplot2 to visualize your findings

Bar chart: Total Spend by Customer Lifestage and Premium Category

customer_segment_summary %>% ggplot(aes(x = LIFESTAGE, y = Total_Spend, fill = PREMIUM_CUSTOMER)) + geom_col(position = "dodge") + labs(title = "Total Spend by Customer

Lifestage and Premium Category", x = "Customer Lifestage", y = "Total Spend ($)") + theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1)) # Rotate x-axis labels

Bar chart: Average Spend Per Customer by Customer Lifestage and Premium Category

customer_segment_summary %>% ggplot(aes(x = LIFESTAGE, y = Avg_Spend_Per_Customer, fill = PREMIUM_CUSTOMER)) + geom_col(position = "dodge") + labs(title = "Average Spend Per Customer by Lifestage and Premium Category", x = "Customer Lifestage", y = "Average Spend Per Customer ($)") + theme_minimal() + theme(axis.text.x = element_text(angle = 45, hjust = 1))

Example: Visualize top brands by a specific segment (e.g., "OLDER FAMILIES" "Mainstream")

brand_by_segment %>% filter(LIFESTAGE == "OLDER FAMILIES", PREMIUM_CUSTOMER == "Mainstream") %>% ggplot(aes(x = reorder(BRAND, Sales), y = Sales)) + geom_col(fill = "purple") + labs(title = "Top Brands for Older Families - Mainstream", x = "Brand", y = "Total Sales ($)") + coord_flip() + theme_minimal()

Add more visualizations as needed based on your specific insights

— 8. Strategic Recommendation Section (Text in R Markdown) —

After your code chunks, you can write your strategic recommendations directly in R Markdown using headings, bullet points, etc.

For example:

Strategic Recommendations for Julia

Based on the analysis of purchasing trends and behaviors, we propose the following strategic recommendations for the upcoming category review:

Key Findings:

* Finding 1: [State your first key finding from the data, e.g., "Older families (Mainstream) represent the highest total spenders in the chips category."]

* Finding 2: [State your second key finding, e.g., "Young Singles/Couples (Budget) show a strong preference for 175g pack sizes of Doritos."]

Actionable Recommendations:

1. Recommendation 1: [Translate Finding 1 into an action, e.g., "Implement targeted promotions for larger pack sizes (e.g., 175g and 330g) specifically aimed at Older Families (Mainstream and Premium segments) through in-store displays and loyalty program offers."]

2. Recommendation 2: [Translate Finding 2 into an action, e.g., "Increase shelf space and promotional activity for Doritos 175g packs in areas frequented by young demographics, particularly near