Quantium Chips Analysis

Priyanshi Singh

2025-06-19

Contents

Introduction

This analysis explores customer segments and chip purchasing behavior to support a strategic recommendation for the Category Manager. We aim to understand which segments purchase the most chips, how much they spend, and their product preferences.

Load and Preview Data

```
# Load data
transactions <- read_csv(here("..", "data", "QVI_transaction_data.csv"))
```

```
## Rows: 264836 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): PROD_NAME
```

```
## dbl (7): DATE, STORE_NBR, LYLTY_CARD_NBR, TXN_ID, PROD_NBR, PROD_QTY, TOT_SALES
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
customers <- read_csv(here("..", "data", "QVI_purchase_behaviour.csv"))
```

```
## Rows: 72637 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): LIFESTAGE, PREMIUM_CUSTOMER
## dbl (1): LYLTY_CARD_NBR
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Preview
glimpse(transactions)
```

```
## Rows: 264,836
## Columns: 8
## $ DATE           <dbl> 43390, 43599, 43605, 43329, 43330, 43604, 43601, 43601,~
## $ STORE_NBR      <dbl> 1, 1, 1, 2, 2, 4, 4, 4, 5, 7, 7, 8, 9, 13, 19, 20, 20, ~
## $ LYLTY_CARD_NBR <dbl> 1000, 1307, 1343, 2373, 2426, 4074, 4149, 4196, 5026, 7~
## $ TXN_ID         <dbl> 1, 348, 383, 974, 1038, 2982, 3333, 3539, 4525, 6900, 7~
## $ PROD_NBR       <dbl> 5, 66, 61, 69, 108, 57, 16, 24, 42, 52, 16, 114, 15, 92~
## $ PROD_NAME      <chr> "Natural Chip        Compny SeaSalt175g", "CCs Nacho Ch~
## $ PROD_QTY       <dbl> 2, 3, 2, 5, 3, 1, 1, 1, 1, 2, 1, 5, 2, 1, 1, 1, 4, 1, 1~
## $ TOT_SALES      <dbl> 6.0, 6.3, 2.9, 15.0, 13.8, 5.1, 5.7, 3.6, 3.9, 7.2, 5.7~
```

```
glimpse(customers)
```

```
## Rows: 72,637
## Columns: 3
## $ LYLTY_CARD_NBR   <dbl> 1000, 1002, 1003, 1004, 1005, 1007, 1009, 1010, 1011,~
## $ LIFESTAGE        <chr> "YOUNG SINGLES/COUPLES", "YOUNG SINGLES/COUPLES", "YO~
## $ PREMIUM_CUSTOMER <chr> "Premium", "Mainstream", "Budget", "Mainstream", "Mai~
```

Data Cleaning

```
# Convert date
transactions$DATE <- as.Date(transactions$DATE)

# Remove outliers (optional threshold)
```

```r
transactions <- transactions %>%
  filter(PROD_QTY > 0, PROD_QTY < 100)

# Check missing values
colSums(is.na(transactions))
```

```
##           DATE      STORE_NBR LYLTY_CARD_NBR         TXN_ID       PROD_NBR
##              0              0              0              0              0
##      PROD_NAME       PROD_QTY      TOT_SALES
##              0              0              0
```

```r
colSums(is.na(customers))
```

```
##   LYLTY_CARD_NBR        LIFESTAGE PREMIUM_CUSTOMER
##                0                0                0
```

Feature Engineering (Brand & Pack Size)

```r
# Brand is the first word in PROD_NAME
transactions <- transactions %>%
  mutate(BRAND = word(PROD_NAME, 1),
         PACK_SIZE = parse_number(PROD_NAME))

# Check top brands
transactions %>% count(BRAND, sort = TRUE)
```

```
## # A tibble: 29 x 2
##    BRAND         n
##    <chr>     <int>
##  1 Kettle    41288
##  2 Smiths    28860
##  3 Pringles  25102
##  4 Doritos   24962
##  5 Thins     14075
##  6 RRD       11894
##  7 Infuzions 11057
##  8 WW        10320
##  9 Cobs       9693
## 10 Tostitos   9471
## # i 19 more rows
```

Merge with Customer Data

```r
data_combined <- merge(transactions, customers, by = "LYLTY_CARD_NBR")
glimpse(data_combined)
```

```
## Rows: 264,834
## Columns: 12
## $ LYLTY_CARD_NBR   <dbl> 1000, 1002, 1003, 1003, 1004, 1005, 1007, 1007, 1009,~
## $ DATE             <date> 2088-10-18, 2088-09-17, 2089-03-08, 2089-03-09, 2088~
## $ STORE_NBR        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
## $ TXN_ID           <dbl> 1, 2, 3, 4, 5, 6, 8, 7, 9, 10, 11, 15, 14, 12, 13, 17~
## $ PROD_NBR         <dbl> 5, 58, 52, 106, 96, 86, 10, 49, 20, 51, 59, 1, 49, 84~
## $ PROD_NAME        <chr> "Natural Chip        Compny SeaSalt175g", "Red Rock D~
## $ PROD_QTY         <dbl> 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 1, 1, 2, 1, 1, 1, 1,~
## $ TOT_SALES        <dbl> 6.0, 2.7, 3.6, 3.0, 1.9, 2.8, 2.7, 3.8, 5.7, 8.8, 5.1~
## $ BRAND            <chr> "Natural", "Red", "Grain", "Natural", "WW", "Cheetos"~
## $ PACK_SIZE        <dbl> 175, 150, 210, 175, 160, 165, 150, 110, 330, 170, 300~
## $ LIFESTAGE        <chr> "YOUNG SINGLES/COUPLES", "YOUNG SINGLES/COUPLES", "YO~
## $ PREMIUM_CUSTOMER <chr> "Premium", "Mainstream", "Budget", "Budget", "Mainstr~
```

Customer Segment Analysis

```r
segment_summary <- data_combined %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(
    total_sales = sum(TOT_SALES),
    avg_price = mean(TOT_SALES / PROD_QTY),
    transaction_count = n()
  ) %>%
  arrange(desc(total_sales))
```

```
## `summarise()` has grouped output by 'LIFESTAGE'. You can override using the
## `.groups` argument.
```

```r
segment_summary
```

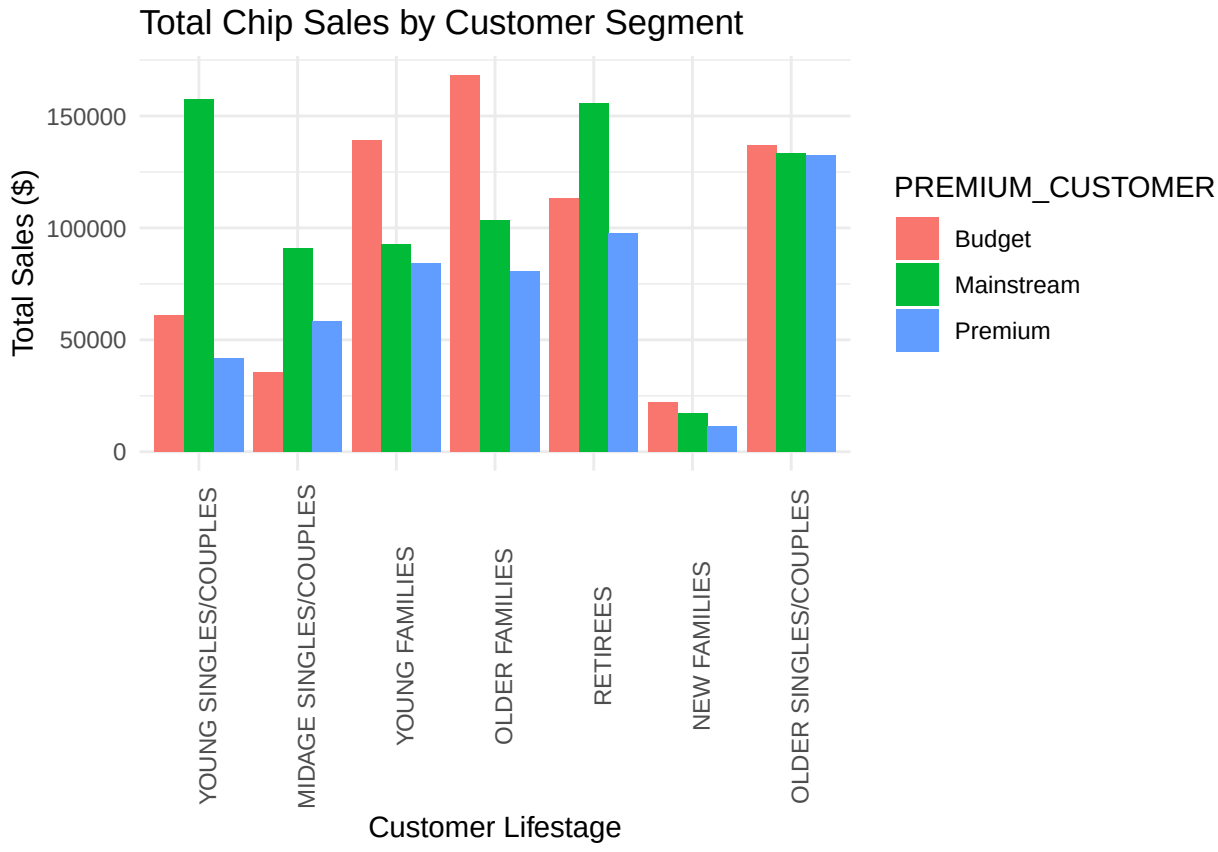
```
## # A tibble: 21 x 5
## # Groups:   LIFESTAGE [7]
##    LIFESTAGE            PREMIUM_CUSTOMER total_sales avg_price transaction_count
##    <chr>                <chr>                  <dbl>     <dbl>             <int>
##  1 OLDER FAMILIES       Budget               168363.      3.73             23160
##  2 YOUNG SINGLES/COUPL~ Mainstream           157622.      4.07             20854
##  3 RETIREES             Mainstream           155677.      3.83             21466
##  4 YOUNG FAMILIES       Budget               139346.      3.75             19122
##  5 OLDER SINGLES/COUPL~ Budget               136770.      3.88             18407
```

```
##  6 OLDER SINGLES/COUPL~ Mainstream           133394.      3.80            18318
##  7 OLDER SINGLES/COUPL~ Premium              132263.      3.89            17754
##  8 RETIREES             Budget               113148.      3.92            15201
##  9 OLDER FAMILIES       Mainstream           103446.      3.73            14244
## 10 RETIREES             Premium               97646.      3.92            13096
## # i 11 more rows
```

```r
segment_summary$LIFESTAGE <- factor(segment_summary$LIFESTAGE, levels = c(
  "YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES", "YOUNG FAMILIES",
  "OLDER FAMILIES", "RETIREES", "NEW FAMILIES", "OLDER SINGLES/COUPLES"
))
```

Visualization: Total Sales by Segment

```r
ggplot(segment_summary, aes(x = LIFESTAGE, y = total_sales, fill = PREMIUM_CUSTOMER)) +
  geom_col(position = "dodge") +
  labs(title = "Total Chip Sales by Customer Segment", y = "Total Sales ($)", x = "Customer Li
  theme_minimal() + theme(axis.text.x = element_text(angle = 90, hjust = 0.5))
```



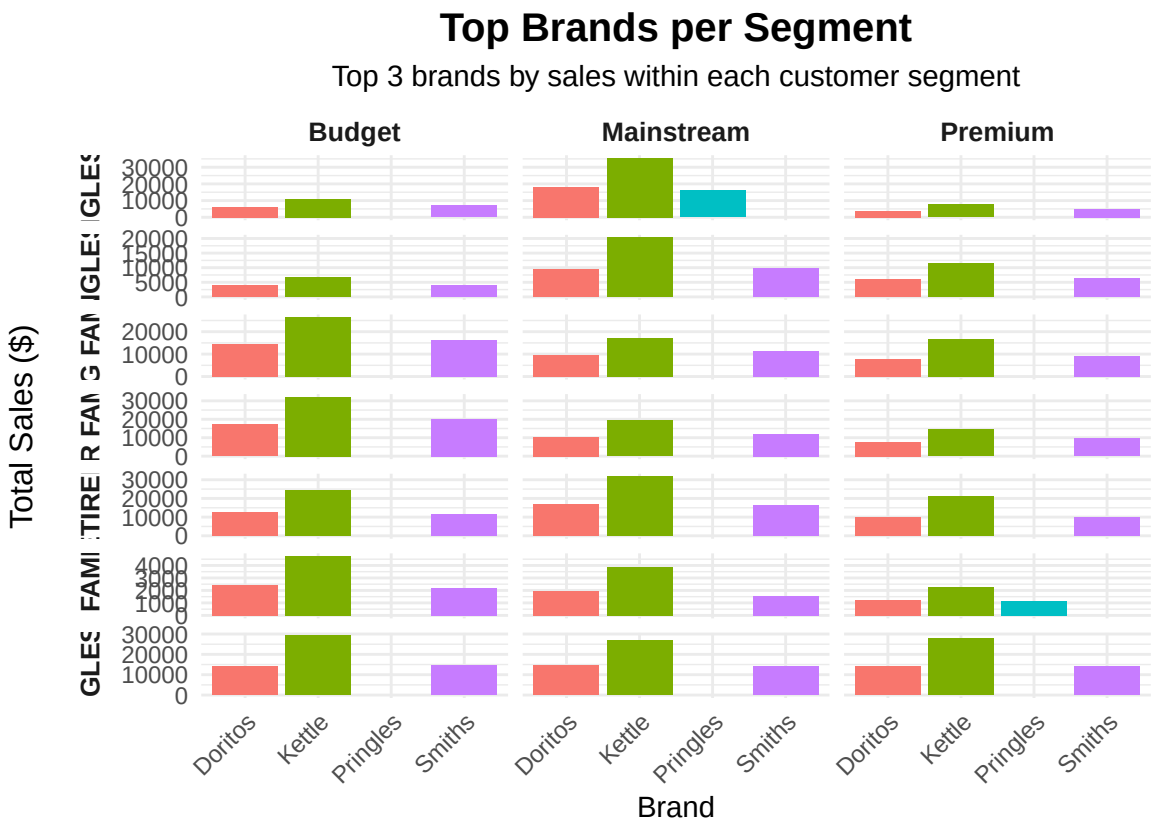
Top Brands by Customer Segment

```r
library(dplyr)
library(ggplot2)

# Create top brands summary
top_brands <- data_combined %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER, BRAND) %>%
  summarise(total_sales = sum(TOT_SALES), .groups = "drop") %>%
  arrange(LIFESTAGE, PREMIUM_CUSTOMER, desc(total_sales)) %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  slice_max(order_by = total_sales, n = 3)

# Fix factor levels
top_brands$LIFESTAGE <- factor(top_brands$LIFESTAGE, levels = c(
  "YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES", "YOUNG FAMILIES",
  "OLDER FAMILIES", "RETIREES", "NEW FAMILIES", "OLDER SINGLES/COUPLES"
))
```

```r
ggplot(top_brands, aes(x = BRAND, y = total_sales, fill = BRAND)) +
  geom_col(show.legend = FALSE) +
  facet_grid(rows = vars(LIFESTAGE), cols = vars(PREMIUM_CUSTOMER), scales = "free_y", switch =
  labs(
    title = "Top Brands per Segment",
    subtitle = "Top 3 brands by sales within each customer segment",
    x = "Brand",
    y = "Total Sales ($)"  #  clear label
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 9),
    axis.text.y = element_text(size = 9),
    strip.text = element_text(face = "bold", size = 10),
    strip.placement = "outside",
    plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
    plot.subtitle = element_text(size = 11, hjust = 0.5),
    axis.title.y = element_text(margin = margin(r = 10), size = 12),  # prevent cutoff
    plot.margin = margin(10, 20, 10, 20)  # extra padding on all sides
  )
```



Pack Size Preferences

```
pack_size_summary <- data_combined %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER, PACK_SIZE) %>%
  summarise(
    total_sales = sum(TOT_SALES),
    transactions = n()
  ) %>%
  arrange(desc(total_sales))
```

```
## `summarise()` has grouped output by 'LIFESTAGE', 'PREMIUM_CUSTOMER'. You can
## override using the `.groups` argument.
```

```
pack_size_summary %>% head(12)
```

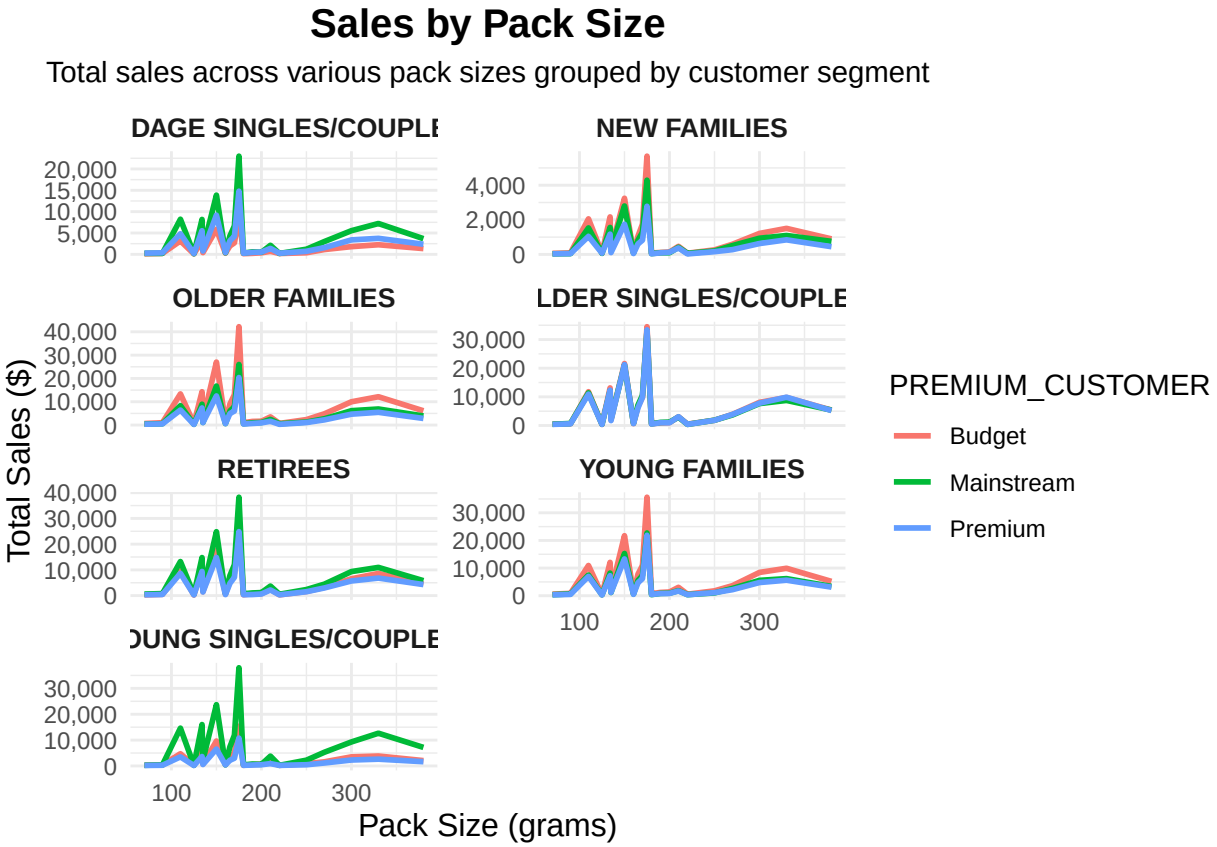
```
## # A tibble: 12 x 5
## # Groups:   LIFESTAGE, PREMIUM_CUSTOMER [10]
##    LIFESTAGE             PREMIUM_CUSTOMER PACK_SIZE total_sales transactions
##    <chr>                 <chr>                <dbl>       <dbl>        <int>
##  1 OLDER FAMILIES        Budget                 175      42205.         5808
```

```
##  2 RETIREES              Mainstream            175      38243.         5295
##  3 YOUNG SINGLES/COUPLES Mainstream            175      37968.         4997
##  4 YOUNG FAMILIES        Budget                175      35635.         4921
##  5 OLDER SINGLES/COUPLES Budget                175      34497          4625
##  6 OLDER SINGLES/COUPLES Premium               175      33393.         4458
##  7 OLDER SINGLES/COUPLES Mainstream            175      33042.         4525
##  8 RETIREES              Budget                175      28977.         3847
##  9 OLDER FAMILIES        Budget                150      27017.         3882
## 10 OLDER FAMILIES        Mainstream            175      25975.         3588
## 11 RETIREES              Premium               175      24868.         3306
## 12 RETIREES              Mainstream            150      24840.         3522
```

```
##  Pack Size Preferences
###  Line chart comparing sales trends across pack sizes in each customer segment

ggplot(pack_size_summary, aes(x = PACK_SIZE, y = total_sales, color = PREMIUM_CUSTOMER)) +
  geom_line(size = 1) +
  facet_wrap(~LIFESTAGE, scales = "free_y", ncol = 2) +
  labs(
    title = "Sales by Pack Size",
    subtitle = "Total sales across various pack sizes grouped by customer segment",
    x = "Pack Size (grams)",
    y = "Total Sales ($)"
  ) +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(
    axis.text = element_text(size = 9),
    axis.title = element_text(size = 12),
    strip.text = element_text(face = "bold", size = 10),
    plot.title = element_text(face = "bold", size = 15, hjust = 0.5),
    plot.subtitle = element_text(size = 11, hjust = 0.5),
    legend.position = "right"
  )
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Strategic Recommendations for Julia

Based on our insights:

- Focus on **Young Singles/Couples** and **Mainstream Midage** as they drive highest sales.
- Promote **larger pack sizes** and premium brands to segments that spend more per transaction.
- Use loyalty program data to promote **top 3 brands** per segment via email offers.
- Avoid low-volume brands for budget-conscious lifestages.

Export Summary Data (Optional)

```
write_csv(segment_summary, "output/segment_summary.csv")
write_csv(pack_size_summary, "output/pack_size_summary.csv")
write_csv(data_combined, "output/cleaned_data.csv")
ggsave("top_brands_per_segment.pdf", width = 10, height = 8)
ggsave("pack_size_plot.pdf", width = 10, height = 8)
```

Conclusion

This report summarizes key insights on customer behavior and chip sales. It enables data-driven targeting of high-value segments, tailoring of product offerings, and strategic promotional planning.