```
pdf("monthly_chip_sales_by_lifestage.pdf", width = 10, height = 6)
# ------------------ 1. Load Packages ------------------
library(readxl)
library(ggplot2)
library(dplyr)

# ------------------ 2. Load Data ------------------
transaction_file <- "QVI_transaction_data.xlsx"
customer_file <- "QVI_purchase_behaviour.csv"

transactions <- read_excel(transaction_file, sheet = 1)
customers <- read.csv(customer_file)

# Convert DATE to proper format
transactions$DATE <- as.Date(transactions$DATE, origin = "1899-12-30")

# ------------------ 3. Merge and Initial Cleaning ------------------
merged_data <- transactions %>%
  inner_join(customers, by = "LYLTY_CARD_NBR")

# Remove non-chip products (e.g. salsa)
merged_data <- merged_data %>%
  filter(!grepl("salsa", PROD_NAME, ignore.case = TRUE))

# Remove extreme purchase quantities (likely non-retail)
merged_data <- merged_data %>%
  filter(PROD_QTY < 50)

# ------------------ 4. Feature Engineering ------------------
# Extract pack size from product name
merged_data$PACK_SIZE <- as.numeric(gsub(".*?(\\d{2,3})[Gg].*", "\\1",
merged_data$PROD_NAME))

# Standardize product names to uppercase
merged_data$PROD_NAME <- toupper(merged_data$PROD_NAME)

# Define known brands (sorted longest to shortest)
known_brands <- c("GRAIN WAVES", "BURGER RINGS", "FRENCH FRIES", "RED ROCK DELI",
"NATURAL CHIP CO",
                  "WOOLWORTHS", "BLACKSTONE", "TYRRELLS", "TOSTITOS", "CHEEZELS",
"PRINGLES", "THINS",
                  "TWISTIES", "INFUZIONS", "DORITOS", "SMITHS", "KETTLE", "CCS",
"COLES", "DELITES",
                  "Cheetos", "COBS", "WW", "RRD", "RED", "NCC", "GRNWVES", "INFZNS",
"POPD", "GRAIN")

known_brands <- known_brands[order(-nchar(known_brands))]  # Longest match first
new_brands <- c("NATURAL CHIP", "NATURAL CHIPCO", "SUNBITES", "SNBTS", "CHEETOS",
"DORITO", "SMITH")
known_brands <- unique(c(known_brands, new_brands))
known_brands <- known_brands[order(-nchar(known_brands))]  # re-sort

# Extract brand name from PROD_NAME
get_brand <- function(name) {
  for (brand in known_brands) {
    if (grepl(brand, name)) return(brand)
  }
  return(NA)
}

merged_data$BRAND_RAW <- sapply(merged_data$PROD_NAME, get_brand)

# Map raw brand names to standardized names
brand_map <- c(
  "RRD" = "Red Rock Deli", "RED" = "Red Rock Deli",
  "NATURAL" = "Natural Chip Co", "NCC" = "Natural Chip Co",
  "GRNWVES" = "Grain Waves", "GRAIN WAVES" = "Grain Waves", "GRAIN" = "Grain Waves",
  "SMITHS" = "Smiths", "SMITH" = "Smiths",
  "WW" = "Woolworths", "WOOLWORTHS" = "Woolworths",
```

```
  "KETTLE" = "Kettle", "INFUZIONS" = "Infuzions", "INFZNS" = "Infuzions",
  "TOSTITOS" = "Tostitos", "CHEEZELS" = "Cheezels",
  "PRINGLES" = "Pringles", "THINS" = "Thins",
  "CCS" = "CCs", "DORITOS" = "Doritos",
  "BLACKSTONE" = "Blackstone", "TYRRELLS" = "Tyrrells",
  "COLES" = "Coles", "DELITES" = "Delites",
  "TWISTIES" = "Twisties", "BURGER RINGS" = "Burger Rings",
  "CHEETOS" = "Cheetos", "FRENCH FRIES" = "French Fries",
  "COBS" = "Cobs", "POPD" = "Cobs","NATURAL CHIP" = "Natural Chip Co",
  "NATURAL CHIPCO" = "Natural Chip Co",
  "DORITO" = "Doritos",
  "SNBTS" = "Sunbites",
  "SUNBITES" = "Sunbites",
  "CHEETOS" = "Cheetos",
  "SMITH" = "Smiths"
)


merged_data$BRAND <- brand_map[merged_data$BRAND_RAW]
merged_data$BRAND[is.na(merged_data$BRAND)] <-
merged_data$BRAND_RAW[is.na(merged_data$BRAND)]

# ------------------ 5. Segment Summary ------------------
segment_summary <- merged_data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(
    total_sales = sum(TOT_SALES),
    avg_quantity = mean(PROD_QTY),
    avg_pack = mean(PACK_SIZE, na.rm = TRUE),
    num_transactions = n(),
    .groups = 'drop'
  ) %>%
  arrange(desc(total_sales))

write.csv(merged_data, "cleaned_chip_data.csv", row.names = FALSE)
write.csv(segment_summary, "segment_summary.csv", row.names = FALSE)

# ------------------ 6. Additional Features ------------------
# Pack groupings
merged_data$PACK_GROUP <- cut(
  merged_data$PACK_SIZE,
  breaks = c(0, 150, 200, Inf),
  labels = c("Small (<150g)", "Medium (150-200g)", "Large (>200g)"),
  right = TRUE
)

# Monthly sales
merged_data$MONTH <- as.Date(format(merged_data$DATE, "%Y-%m-01"))
# ------------------ 7. -tests ------------------

# Unit price
merged_data$UNIT_PRICE <- merged_data$TOT_SALES / merged_data$PROD_QTY

test_data <- merged_data %>%
  filter(LIFESTAGE == "YOUNG SINGLES/COUPLES",
         PREMIUM_CUSTOMER %in% c("Mainstream", "Premium"))
t_test_result <- t.test(
  UNIT_PRICE ~ PREMIUM_CUSTOMER,  # compare unit price by group
  data = test_data,
  var.equal = FALSE               # Welch's t-test (default)
)

print(t_test_result)
#chi-squared test on pack size vs customer group
table_data <- table(merged_data$PREMIUM_CUSTOMER, merged_data$PACK_GROUP)
chisq.test(table_data)
#ANOVA
anova_result <- aov(UNIT_PRICE ~ LIFESTAGE, data = merged_data)
summary(anova_result)
```

```
TukeyHSD(anova_result)

#correlation analysis:
cor.test(merged_data$PACK_SIZE, merged_data$UNIT_PRICE, use = "complete.obs")
#cost per gram
merged_data$PRICE_PER_GRAM <- merged_data$UNIT_PRICE / merged_data$PACK_SIZE
cor.test(merged_data$PACK_SIZE, merged_data$PRICE_PER_GRAM, use = "complete.obs")

#proportion test
# Example: customers buying Large packs
premium_large <- sum(merged_data$PREMIUM_CUSTOMER == "Premium" & merged_data$PACK_GROUP
== "Large (>200g)")
premium_total <- sum(merged_data$PREMIUM_CUSTOMER == "Premium")

mainstream_large <- sum(merged_data$PREMIUM_CUSTOMER == "Mainstream" &
merged_data$PACK_GROUP == "Large (>200g)")
mainstream_total <- sum(merged_data$PREMIUM_CUSTOMER == "Mainstream")

prop.test(c(premium_large, mainstream_large), c(premium_total, mainstream_total))




# ------------------ 8. Plots ------------------
# Total Sales by Customer Segment
segment_summary$LIFESTAGE <- with(segment_summary, reorder(LIFESTAGE, total_sales))
ggplot(segment_summary, aes(x = LIFESTAGE, y = total_sales, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  coord_flip() +
  scale_fill_manual(values = c(
    "Budget" = "#59085a",
    "Mainstream" = "#9b59b6",
    "Premium" = "#d2b4de"
  ))+

  labs(title = "Total Sales by Customer Segment", x = "Customer Lifestage", y = "Total
Sales ($)", fill = "Customer Type") +
  theme_minimal()

# Transactions by Customer Segment
ggplot(segment_summary, aes(x = LIFESTAGE, y = num_transactions, fill =
PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  coord_flip() +
  scale_fill_manual(values = c(
    "Budget" = "#59085a",
    "Mainstream" = "#9b59b6",
    "Premium" = "#d2b4de"
  ))+

  labs(title = "Transactions by Customer Segment", x = "Lifestage", y = "Number of
Transactions") +
  theme_minimal()

# Pack Size Preference by Customer Lifestage
pack_segment <- merged_data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER, PACK_GROUP) %>%
  summarise(count = n(), .groups = 'drop')

ggplot(pack_segment, aes(x = PACK_GROUP, y = count, fill = LIFESTAGE)) +
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c(
    "YOUNG SINGLES/COUPLES" = "#59085a",
    "YOUNG FAMILIES" = "#722975",
    "MIDAGE SINGLES/COUPLES" = "#9b59b6",
    "NEW FAMILIES" = "#b37fd9",
    "OLDER SINGLES/COUPLES" = "#c39bd3",
    "OLDER FAMILIES" = "#d2b4de",
    "RETIREES" = "#ebdef0"
```

```
  )) +
  labs(
    title = "Pack Size Preference by Customer Lifestage",
    x = "Pack Size Group",
    y = "Number of Purchases",
    fill = "Lifestage"
  ) +
  theme_minimal()


# Top Brands per Customer Lifestage
top_brands <- merged_data %>%
  group_by(LIFESTAGE, BRAND) %>%
  summarise(purchases = n(), .groups = "drop") %>%
  group_by(LIFESTAGE) %>%
  slice_max(purchases, n = 10)

ggplot(top_brands, aes(x = reorder(BRAND, purchases), y = purchases, fill = LIFESTAGE))
+
  geom_col(show.legend = FALSE) +
  facet_wrap(~ LIFESTAGE, scales = "free") +
  coord_flip() +
  scale_fill_manual(values = c(
    "YOUNG SINGLES/COUPLES" = "#59085a",
    "YOUNG FAMILIES" = "#722975",
    "MIDAGE SINGLES/COUPLES" = "#9b59b6",
    "NEW FAMILIES" = "#b37fd9",
    "OLDER SINGLES/COUPLES" = "#c39bd3",
    "OLDER FAMILIES" = "#d2b4de",
    "RETIREES" = "#ebdef0"
  ))+

  labs(title = "Top 10 Brands per Customer Lifestage", y = "Purchase Count", x =
"Brand")

# Monthly Chip Sales by Lifestage
monthly_sales <- merged_data %>%
  group_by(MONTH, LIFESTAGE) %>%
  summarise(total_sales = sum(TOT_SALES), .groups = "drop")

ggplot(monthly_sales, aes(x = MONTH, y = total_sales, color = LIFESTAGE)) +
  geom_line(size = 1) +
  scale_x_date(date_labels = "%b %Y", date_breaks = "2 months") +  # spacing every 2
months
  scale_color_manual(values = c(
    "YOUNG SINGLES/COUPLES" = "#59085a",
    "YOUNG FAMILIES" = "#722975",
    "MIDAGE SINGLES/COUPLES" = "#9b59b6",
    "NEW FAMILIES" = "#b37fd9",
    "OLDER SINGLES/COUPLES" = "#c39bd3",
    "OLDER FAMILIES" = "#d2b4de",
    "RETIREES" = "#ebdef0"
  )) +
  labs(
    title = "Monthly Chip Sales by Lifestage",
    x = "Month",
    y = "Total Sales ($)",
    color = "Lifestage"
  ) +
  theme_minimal() +
  theme(
    axis.text.x = element_text(angle = 45, hjust = 1, size = 8),  # smaller font
    plot.title = element_text(size = 14, face = "bold")
  )


# Repeat Purchases by Lifestage
customer_freq <- merged_data %>%
  group_by(LYLTY_CARD_NBR, LIFESTAGE) %>%
```

```
  summarise(transactions = n(), .groups = "drop")

ggplot(customer_freq, aes(x = transactions)) +
  geom_histogram(binwidth = 1, fill = "#9b59b6") +
  facet_wrap(~ LIFESTAGE, scales = "free_y") +
  scale_x_continuous(breaks = seq(0, 20, by = 5)) +
  labs(title = "Distribution of Chip Purchases per Customer", x = "Number of Chip
Purchases", y = "Number of Customers") +
  theme_minimal()

# ------------------ 8. Check for Unmatched Brands ------------------
unmatched <- merged_data %>% filter(is.na(BRAND_RAW)) %>% select(PROD_NAME) %>%
unique()
head(unmatched, 20)

# ------------------ End of Script ------------------
dev.off()
getwd()
```