customers and behaviors on chips

KOBENAN_ZAKARI

2025-05-30

```
#Load library
library(readr)
library(data.table)
library(dplyr)
```

```
##
## Attachement du package : 'dplyr'
```

```
## Les objets suivants sont masqués depuis 'package:data.table':
##
##     between, first, last
```

```
## Les objets suivants sont masqués depuis 'package:stats':
##
##     filter, lag
```

```
## Les objets suivants sont masqués depuis 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tidyr)
library(stringr)
library(ggmosaic)
```

```
#TRANSACTION DATA
QVI_transaction_data <- read.csv("QVI_transaction_data.csv", sep = ";")
 str(QVI_transaction_data)
```

```
## 'data.frame':    264836 obs. of  8 variables:
##  $ DATE          : int  43390 43599 43605 43329 43330 43604 43601 43601 43332 43330 ...
##  $ STORE_NBR     : int  1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: int  1000 1307 1343 2373 2426 4074 4149 4196 5026 7150 ...
##  $ TXN_ID        : int  1 348 383 974 1038 2982 3333 3539 4525 6900 ...
##  $ PROD_NBR      : int  5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr  "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g" "Smiths C
##  $ PROD_QTY      : int  2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : chr  "6" "6,3" "2,9" "15" ...
```

```
head(QVI_transaction_data)
```

```
##    DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 1 43390         1           1000      1        5
## 2 43599         1           1307    348       66
## 3 43605         1           1343    383       61
## 4 43329         2           2373    974       69
## 5 43330         2           2426   1038      108
## 6 43604         4           4074   2982       57
##                                 PROD_NAME PROD_QTY TOT_SALES
## 1   Natural Chip        Compny SeaSalt175g        2         6
## 2                 CCs Nacho Cheese    175g        3       6,3
## 3   Smiths Crinkle Cut  Chips Chicken 170g        2       2,9
## 4   Smiths Chip Thinly  S/Cream&Onion 175g        5        15
## 5 Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13,8
## 6 Old El Paso Salsa   Dip Tomato Mild 300g        1       5,1
```

```
summary(QVI_transaction_data)
```

```
##       DATE         STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :43282   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:43373   1st Qu.: 70.0   1st Qu.:  70021   1st Qu.:  67602
##  Median :43464   Median :130.0   Median : 130358   Median : 135138
##  Mean   :43464   Mean   :135.1   Mean   : 135550   Mean   : 135158
##  3rd Qu.:43555   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202701
##  Max.   :43646   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY        TOT_SALES
##  Min.   :  1.00   Length:264836      Min.   :  1.000   Length:264836
##  1st Qu.: 28.00   Class :character   1st Qu.:  2.000   Class :character
##  Median : 56.00   Mode  :character   Median :  2.000   Mode  :character
##  Mean   : 56.58                      Mean   :  1.907
##  3rd Qu.: 85.00                      3rd Qu.:  2.000
##  Max.   :114.00                      Max.   :200.000
```

```
sapply("QVI_transaction_data.csv", class)
```

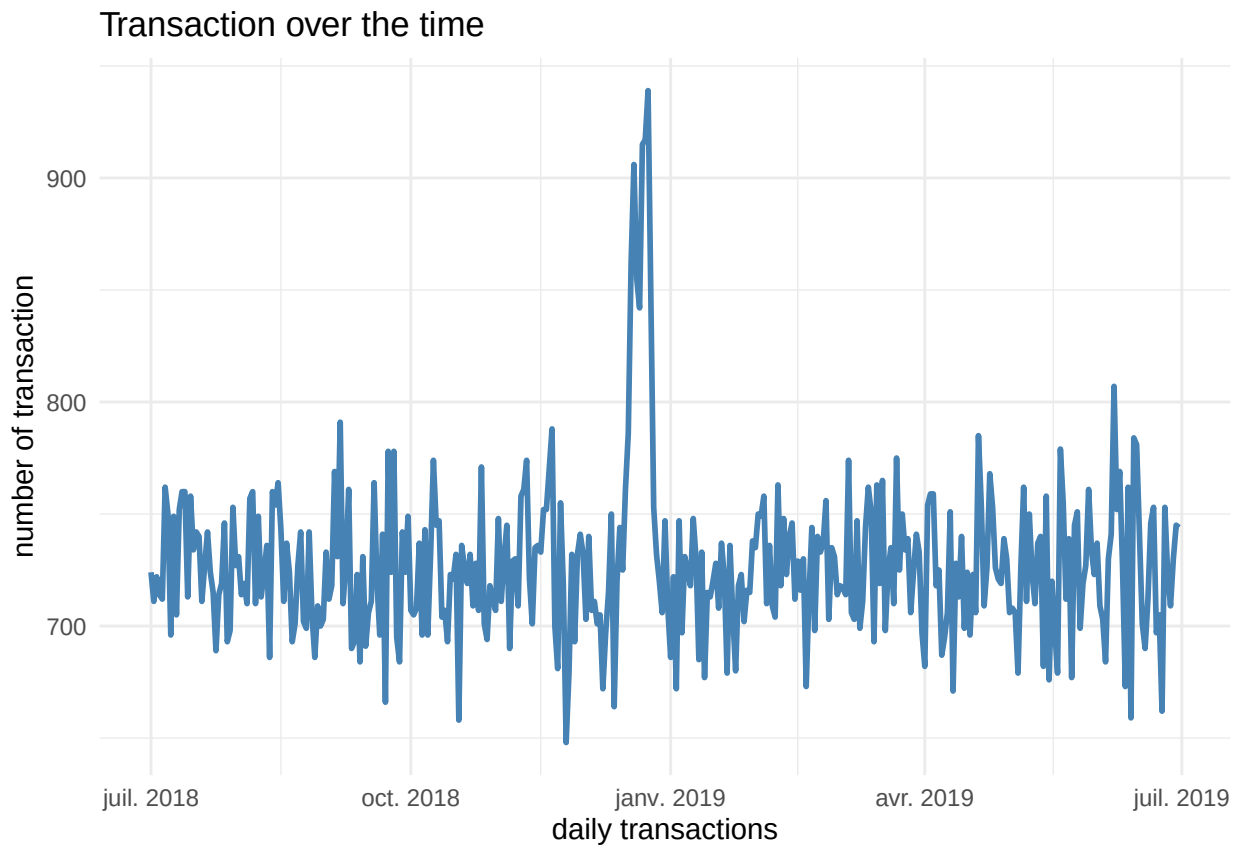
```
## QVI_transaction_data.csv
##              "character"
```

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

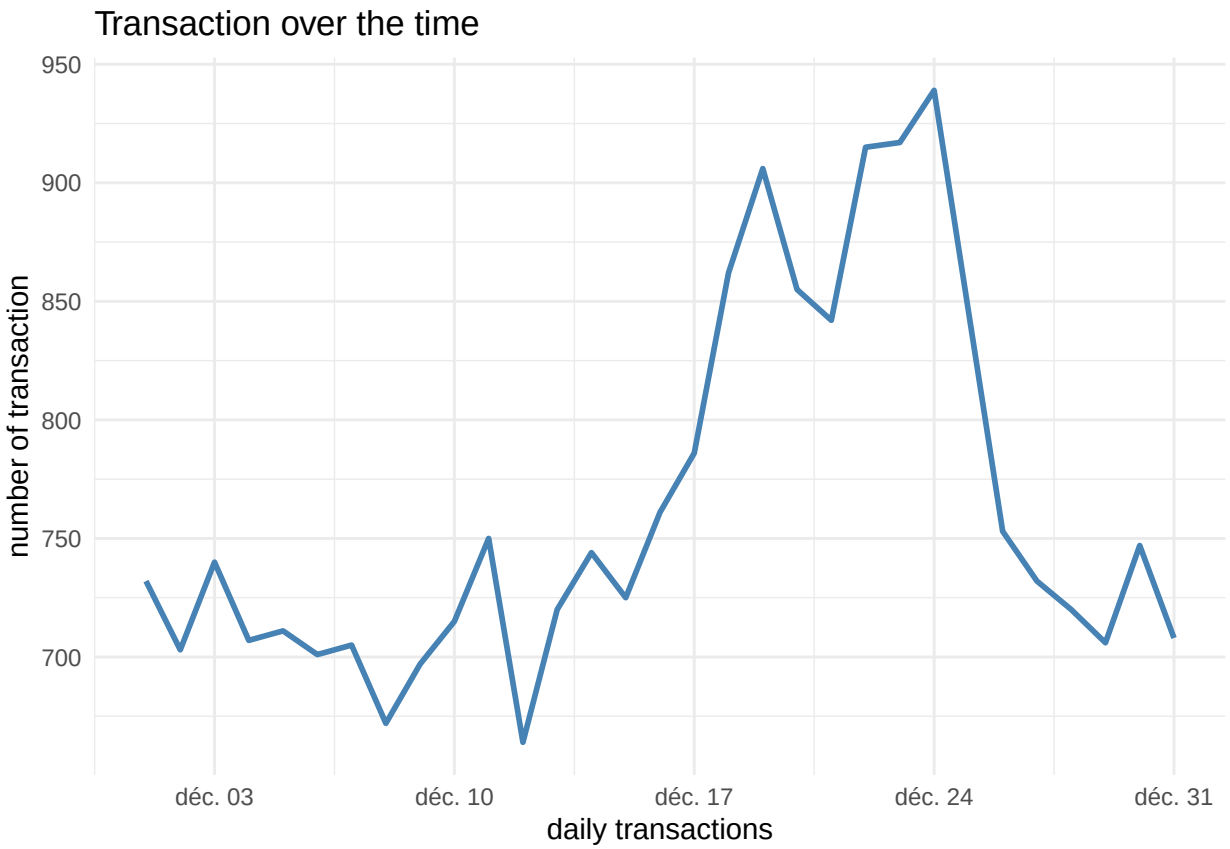
```
## [1] "2018-10-17" "2019-05-14" "2019-05-20" "2018-08-17" "2018-08-18"
## [6] "2019-05-19"
```

```
over_time_transacton <- QVI_transaction_data %>%
  group_by(DATE) %>%
  summarise(transaction_per_date = n())
```

```
ggplot(over_time_transacton, aes(x = DATE,
                                 y= transaction_per_date))+
  geom_line( linewidth = 1,
             color = "steelblue")+
  labs( x = "daily transactions",
        y= "number of transaction",
        title = "Transaction over the time")+
  theme_minimal()
```




```
   # Between oct 2018 and janv 2019, it has been observed an height total of transaction. let deep this
abnormal_period <- QVI_transaction_data %>%
  filter(DATE >= as.Date("2018-12-01") & DATE<= as.Date("2018-12-31")) %>%
  group_by(DATE) %>%
  summarise(count = n())

ggplot(abnormal_period, aes(x = DATE,
                        y= count))+
  geom_line( linewidth = 1,
             color = "steelblue")+
  labs( x = "daily transactions",
        y= "number of transaction",
        title = "Transaction over the time")+
  theme_minimal()
```




```
## from the graph, the over bulk reached on 24 december, the eve of Christmas date and suddlenly has fa
```

```
# Fix identifier structures
identifiers <- QVI_transaction_data[,c("STORE_NBR", "LYLTY_CARD_NBR", "TXN_ID", "PROD_NBR")]
na_identifiers <- colSums (is.na(identifiers))
fix_id_digits <- sapply( identifiers, function(x) sum(grepl("^\\d+$",x)))
```

```
## Fix product names (separation)
QVI_transaction_data <- QVI_transaction_data %>%
  mutate(
    PROD_NAME = gsub("&", " ", PROD_NAME),
    PROD_NAME = gsub("\\s+", " ", PROD_NAME),
    PROD_NAME = gsub("(\\d+)G", " \\1g", PROD_NAME),
    PROD_NAME = gsub("(\\d+g)", " \\1", PROD_NAME),
    PROD_NAME = str_replace_all(PROD_NAME, "(?<=[a-z]) (?=[A-Z])", " ")
  ) %>%
  separate(PROD_NAME, into = c("PROD_NAME", "PROD_SIZE"), sep = " (?=\\d+g)", extra = "merge", fill = "
  mutate(
    PROD_NAME = ifelse(PROD_NAME == "Kettle" & grepl("Swt Pot Sea Salt", PROD_SIZE),
                       paste(PROD_NAME, "Swt Pot Sea Salt"), PROD_NAME),
    PROD_SIZE = gsub("Swt Pot Sea Salt", "", PROD_SIZE)
  )
```

```r
## depend variable (total sales)
QVI_transaction_data$TOT_SALES <- gsub(",", ".", QVI_transaction_data$TOT_SALES)
 QVI_transaction_data$TOT_SALES <- as.numeric(QVI_transaction_data$TOT_SALES)
 head (QVI_transaction_data$TOT_SALES)
```

```
## [1]  6.0  6.3  2.9 15.0 13.8  5.1
```

```r
 Q1 <- quantile (QVI_transaction_data$TOT_SALES, 0.25,na.rm = TRUE)
 Q3 <- quantile (QVI_transaction_data$TOT_SALES, 0.75,na.rm = TRUE)
 IQR <- Q3 - Q1
 lower_bound <- Q1 - 3* IQR
 upper_bound <- Q3 + 3* IQR
 outliers <- QVI_transaction_data$TOT_SALES[QVI_transaction_data$TOT_SALES < lower_bound|QVI_transactio
```

```r
boxplot(outliers,
         main ="Sales outliers",
         ylab= "Total Sales",
         col = "lightcoral",
         outline = TRUE)
```


```r
## Regardless to IQR-based method, it is shown that there are two over sales (depend variable) as "650"
  sales_rows <- QVI_transaction_data [QVI_transaction_data$TOT_SALES== 650,]
```

If the purchases: involve the same customer and product, but occur in different years or seasons, the different TXN_ID values make perfect sense this typpically happens because: each transactions is uniquely logged

even if repeated by the same customer, seasons or year change so the system sees it as a new, the over sales are changed with median is applied :

```
median_sales <- median(QVI_transaction_data$TOT_SALES)
  treshold <- quantile (QVI_transaction_data$TOT_SALES, 0.95)
  print(treshold)
```

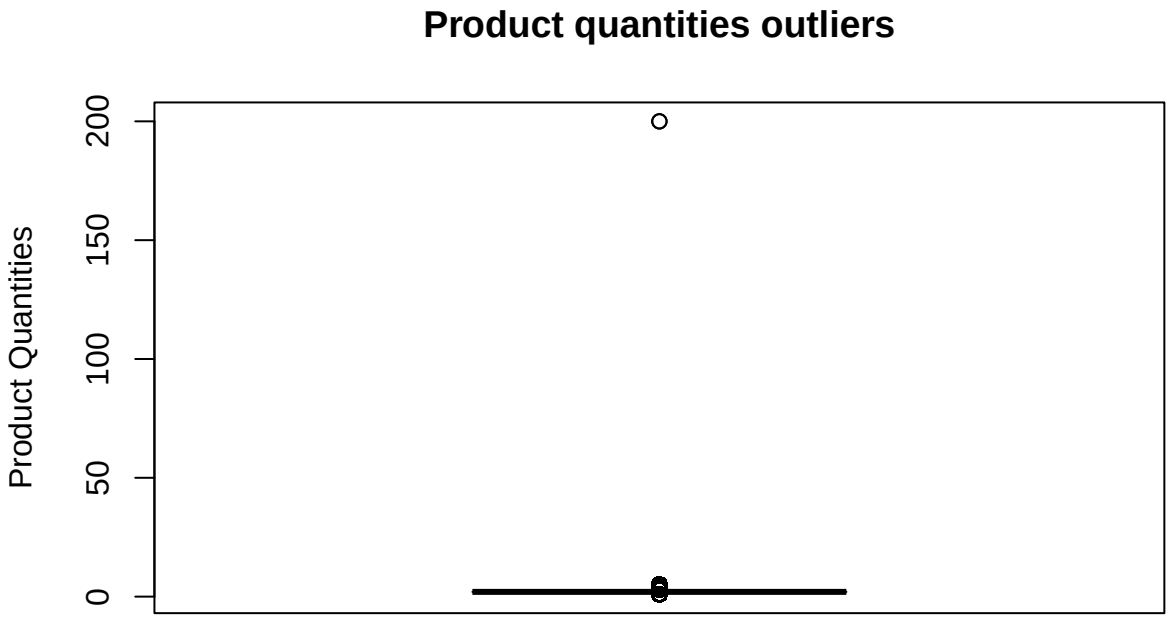
```
##  95%
## 11.4
```

```
  QVI_transaction_data <- QVI_transaction_data %>% mutate(TOT_SALES=ifelse (TOT_SALES> treshold, median
```

```
## Fix Product quantity
   QVI_transaction_data$PROD_QTY <- as.numeric(QVI_transaction_data$PROD_QTY)
   Q1 <- quantile (QVI_transaction_data$PROD_QTY, 0.25,na.rm = TRUE)
   Q3 <- quantile (QVI_transaction_data$PROD_QTY, 0.75,na.rm = TRUE)
   IQR <- Q3 - Q1
   lower_bound <- Q1 - 3* IQR
   upper_bound <- Q3 + 3* IQR
   outliers <- QVI_transaction_data$PROD_QTY[QVI_transaction_data$PROD_QTY < lower_bound|QVI_transactio
```

Changing with median has been made because that include de the same customer

```
boxplot(QVI_transaction_data$PROD_QTY,
            main ="Product quantities outliers",
            ylab = "Product Quantities",
            col = "lightcoral",
            outline = TRUE)
```




```
QTY_rows <- QVI_transaction_data [QVI_transaction_data$PROD_QTY== 200,]
  median_sales <- median(QVI_transaction_data$PROD_QTY)
  treshold <- quantile (QVI_transaction_data$PROD_QTY, 0.95)
  print(treshold)
```

```
## 95%
##   2
```

```
  QVI_transaction_data <- QVI_transaction_data %>% mutate(PROD_QTY=ifelse (PROD_QTY> treshold, median_s
```

```
#CUSTOMER BEHAVIOR DATA
QVI_purchase_behaviour <- read.csv("QVI_purchase_behaviour (1).csv",sep = ",")
head(QVI_purchase_behaviour)
```

```
##   LYLTY_CARD_NBR               LIFESTAGE PREMIUM_CUSTOMER
## 1           1000  YOUNG SINGLES/COUPLES          Premium
## 2           1002  YOUNG SINGLES/COUPLES       Mainstream
## 3           1003         YOUNG FAMILIES           Budget
## 4           1004  OLDER SINGLES/COUPLES       Mainstream
## 5           1005 MIDAGE SINGLES/COUPLES       Mainstream
## 6           1007  YOUNG SINGLES/COUPLES           Budget
```

```r
behaviour_outliers <- sapply(QVI_purchase_behaviour, function(x)sum(is.na(x)))
```

```r
 # DATA COMBINATION
data <- merge(QVI_transaction_data, QVI_purchase_behaviour, by= "LYLTY_CARD_NBR", all.x = TRUE)
head(data)
```

```
##   LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR
## 1           1000 2018-10-17         1      1        5
## 2           1002 2018-09-16         1      2       58
## 3           1003 2019-03-07         1      3       52
## 4           1003 2019-03-08         1      4      106
## 5           1004 2018-11-02         1      5       96
## 6           1005 2018-12-28         1      6       86
##                           PROD_NAME PROD_SIZE PROD_QTY TOT_SALES
## 1       Natural Chip Compny SeaSalt      175g        2       6.0
## 2 Red Rock Deli Chikn Garlic Aioli       150g        1       2.7
## 3   Grain Waves Sour Cream Chives        210g        1       3.6
## 4     Natural ChipCo Hony Soy Chckn      175g        1       3.0
## 5        WW Original Stacked Chips       160g        1       1.9
## 6                     Cheetos Puffs       165g        1       2.8
##                LIFESTAGE PREMIUM_CUSTOMER
## 1  YOUNG SINGLES/COUPLES          Premium
## 2  YOUNG SINGLES/COUPLES       Mainstream
## 3         YOUNG FAMILIES           Budget
## 4         YOUNG FAMILIES           Budget
## 5  OLDER SINGLES/COUPLES       Mainstream
## 6 MIDAGE SINGLES/COUPLES       Mainstream
```

```r
   # DATA EXPLORATION

 # Extract product brands
keywords <- c(
  "chip","chips","snack","snacks","smokey","smocked","grilled", "roasted", "crisp", "salt", "BBQ", "chee
  "onion", "jalapeno", "sea salt", "vinegar","cheddar","parmesan","blue cheese","mozzarella", "nacho",
  "chilli", "hot", "corn", "original", "tangy", "spicy", "Original", "Chicken", "Rings","Supreme","Lime
  "spicy","hot","pepper","mustard","pickle","honey","caramel","chocolate","herb","plantain","wasabi","k
)
data <- data %>%
  mutate(PROD_NAME = tolower(PROD_NAME)) %>%
  filter(grepl(paste(keywords,collapse = "|"),PROD_NAME)) %>%
  separate(PROD_NAME, into = c("BRANDS", "CATEGORY&FLAVOR"), sep = " ", extra= "merge",fill = "right") %
  mutate(BRANDS = toupper (BRANDS))
```

```r
data_brand <- data %>% count(BRANDS, sort = TRUE)

data <- data %>%
  mutate(
    BRANDS = gsub("\\bDORITO\\b", "DORITOS", BRANDS),
    BRANDS = gsub("\\bINFZNS\\b", "INFUZIONS", BRANDS),
    BRANDS = gsub("\\bSMITH\\b", "SMITHS", BRANDS),
    BRANDS = gsub("\\bSNBTS\\b","SUNBITES", BRANDS)
  )
```

```r
data <- data %>%
  mutate(PROD_SIZE = as.numeric(gsub("[^0-9]","",PROD_SIZE)))
head (data)
```

```
##   LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR    BRANDS
## 1           1000 2018-10-17         1      1        5   NATURAL
## 2           1003 2019-03-07         1      3       52     GRAIN
## 3           1003 2019-03-08         1      4      106   NATURAL
## 4           1004 2018-11-02         1      5       96        WW
## 5           1007 2018-12-04         1      7       49 INFUZIONS
## 6           1009 2018-11-20         1      9       20   DORITOS
##              CATEGORY&FLAVOR PROD_SIZE PROD_QTY TOT_SALES            LIFESTAGE
## 1        chip compny seasalt       175        2       6.0 YOUNG SINGLES/COUPLES
## 2  waves sour cream chives         210        1       3.6        YOUNG FAMILIES
## 3      chipco hony soy chckn       175        1       3.0        YOUNG FAMILIES
## 4    original stacked chips        160        1       1.9 OLDER SINGLES/COUPLES
## 5 sourcream herbs veg strws        110        1       3.8 YOUNG SINGLES/COUPLES
## 6            cheese supreme        330        1       5.7          NEW FAMILIES
##   PREMIUM_CUSTOMER
## 1          Premium
## 2           Budget
## 3           Budget
## 4       Mainstream
## 5           Budget
## 6          Premium
```

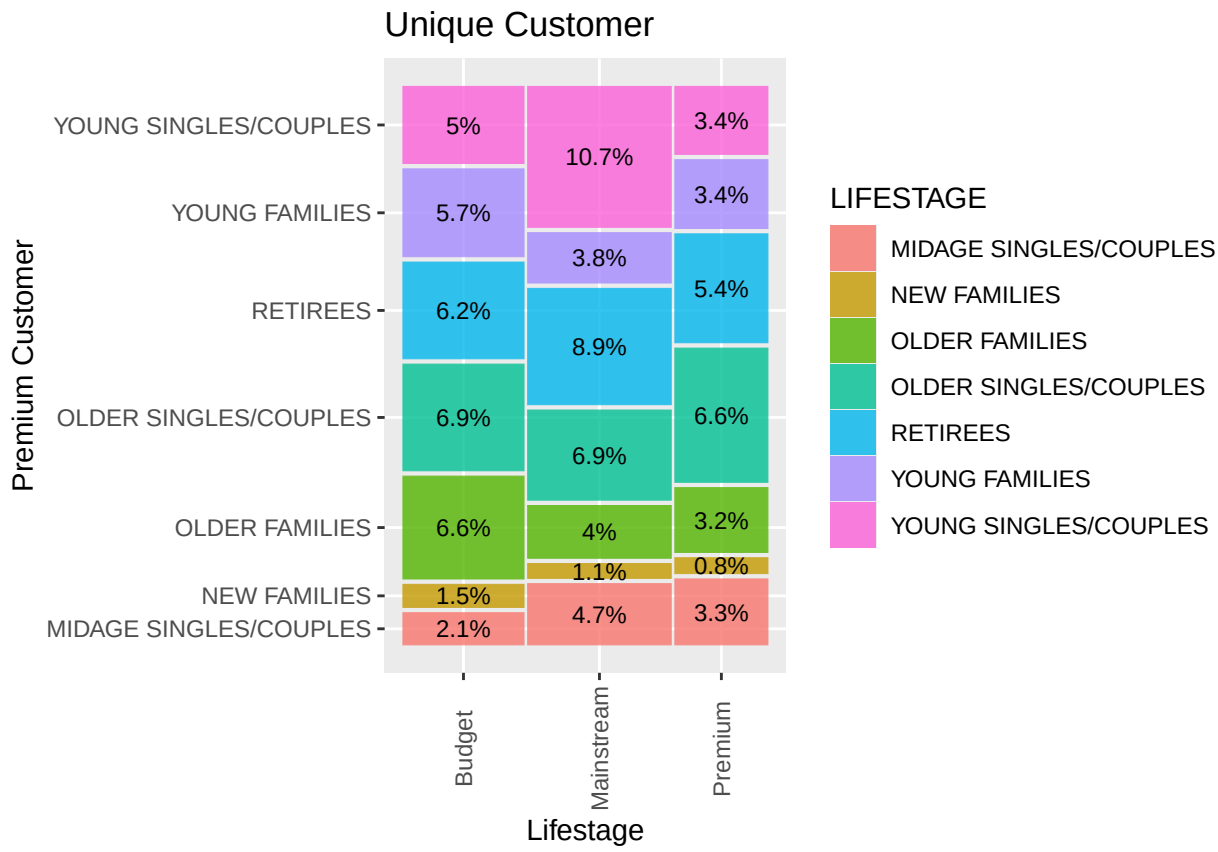
```r
# Understanding of behavior in category

number_unique_customer <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(number_unique_customer = uniqueN(LYLTY_CARD_NBR), .groups = "drop" )

p <- ggplot(data =number_unique_customer ) +
  geom_mosaic(aes(weight = number_unique_customer ,
                  x = product(LIFESTAGE, PREMIUM_CUSTOMER),
                  fill = LIFESTAGE)) +
  labs(x = "Lifestage",
       y = "Premium Customer",
       title = "Unique Customer") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
built_data <- ggplot_build(p)$data[[1]]
```

```
## Warning: The ‘scale_name‘ argument of ‘continuous_scale()‘ is deprecated as of ggplot2
## 3.5.0.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: The ‘trans‘ argument of ‘continuous_scale()‘ is deprecated as of ggplot2 3.5.0.
## i Please use the ‘transform‘ argument instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## i Please use `unite()` instead.
## i The deprecated feature was likely used in the ggmosaic package.
##   Please report the issue at <https://github.com/haleyjeppson/ggmosaic>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

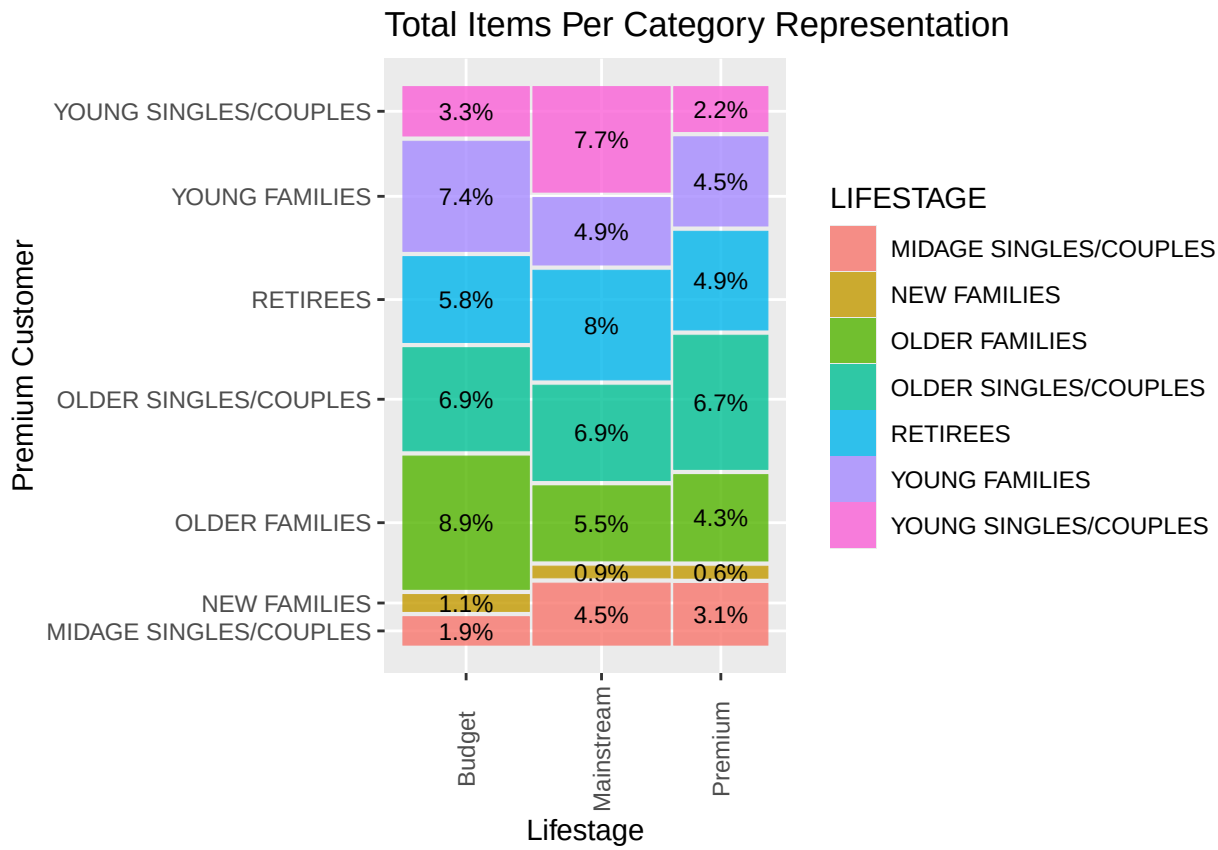
```
p + geom_text(
  data = built_data,
  aes(
    x = (xmin + xmax) / 2,
    y = (ymin + ymax) / 2,
    label = paste0(round(.wt / sum(.wt) * 100, 1), "%")
  ),
  inherit.aes = FALSE,
  size = 3
)
```



Unique Customer (number of customer per group) has shown that the Mainstrean/Retirees makes more transactions that others groups. let's keep checking. the retirees are likely not interested about Chips but just to spend time.

```
Total_items_customer<- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(Total_items_customer = sum (PROD_QTY), .groups = "drop" )
```

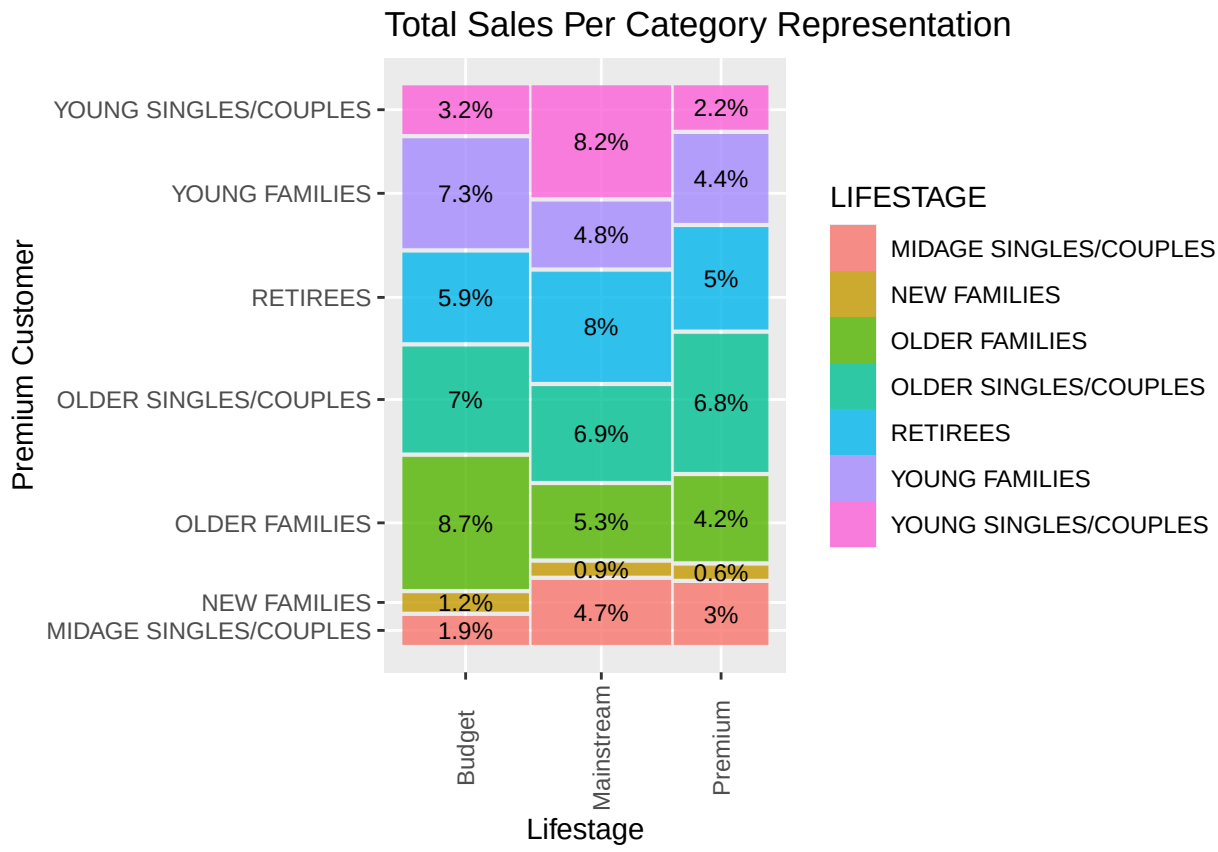
```
p <- ggplot(data =Total_items_customer ) +
  geom_mosaic(aes(weight = Total_items_customer,
                  x = product(LIFESTAGE, PREMIUM_CUSTOMER),
                  fill = LIFESTAGE)) +
  labs(x = "Lifestage",
       y = "Premium Customer",
       title = "Total Items Per Category Representation") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
built_data <- ggplot_build(p)$data[[1]]
p + geom_text(
  data = built_data,
  aes(
    x = (xmin + xmax) / 2,
    y = (ymin + ymax) / 2,
    label = paste0(round(.wt / sum(.wt) * 100, 1), "%")
  ),
  inherit.aes = FALSE,
  size = 3
)
```



Opposite to the percentage of transaction, Budget and Older families set a record among other groups. That make sense because they likely bought it for their children.

```
Total_sales <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(Total_sales = sum (TOT_SALES), .groups = "drop" )
```

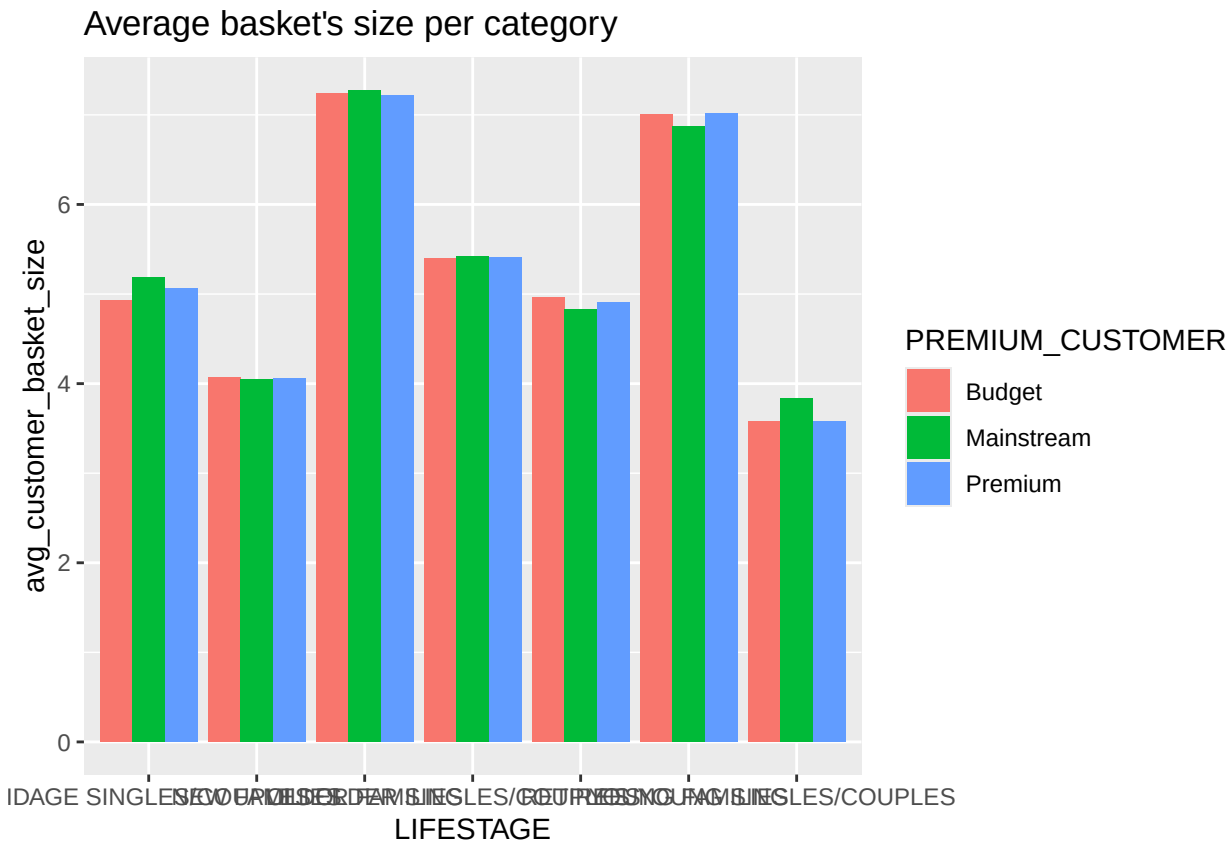
```
p <- ggplot(data = Total_sales) +
  geom_mosaic(aes(weight = Total_sales ,
                  x = product(LIFESTAGE, PREMIUM_CUSTOMER),
                  fill = LIFESTAGE)) +
  labs(x = "Lifestage",
       y = "Premium Customer",
       title = "Total Sales Per Category Representation") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
built_data <- ggplot_build(p)$data[[1]]
p + geom_text(
  data = built_data,
  aes(
    x = (xmin + xmax) / 2,
    y = (ymin + ymax) / 2,
    label = paste0(round(.wt / sum(.wt) * 100, 1), "%")
  ),
  inherit.aes = FALSE,
  size = 3
)
```



The older families and budget big quantity transactions makes them buy more per group. now on, we are going to check global behavior per group.

```
## per category purchase behavior understanding
# how many do they buy per product (quantity)?
avg_customer_basket <- data %>%
```

```
  group_by(LIFESTAGE,PREMIUM_CUSTOMER) %>%
  summarise( avg_customer_basket = sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR), .groups = "drop")

ggplot(avg_customer_basket, aes(weight = avg_customer_basket, x = LIFESTAGE,
                                fill = PREMIUM_CUSTOMER))+
  geom_bar(position = position_dodge())+
  labs (x = "LIFESTAGE", y = "avg_customer_basket_size", title = "Average basket's size per category " )
```



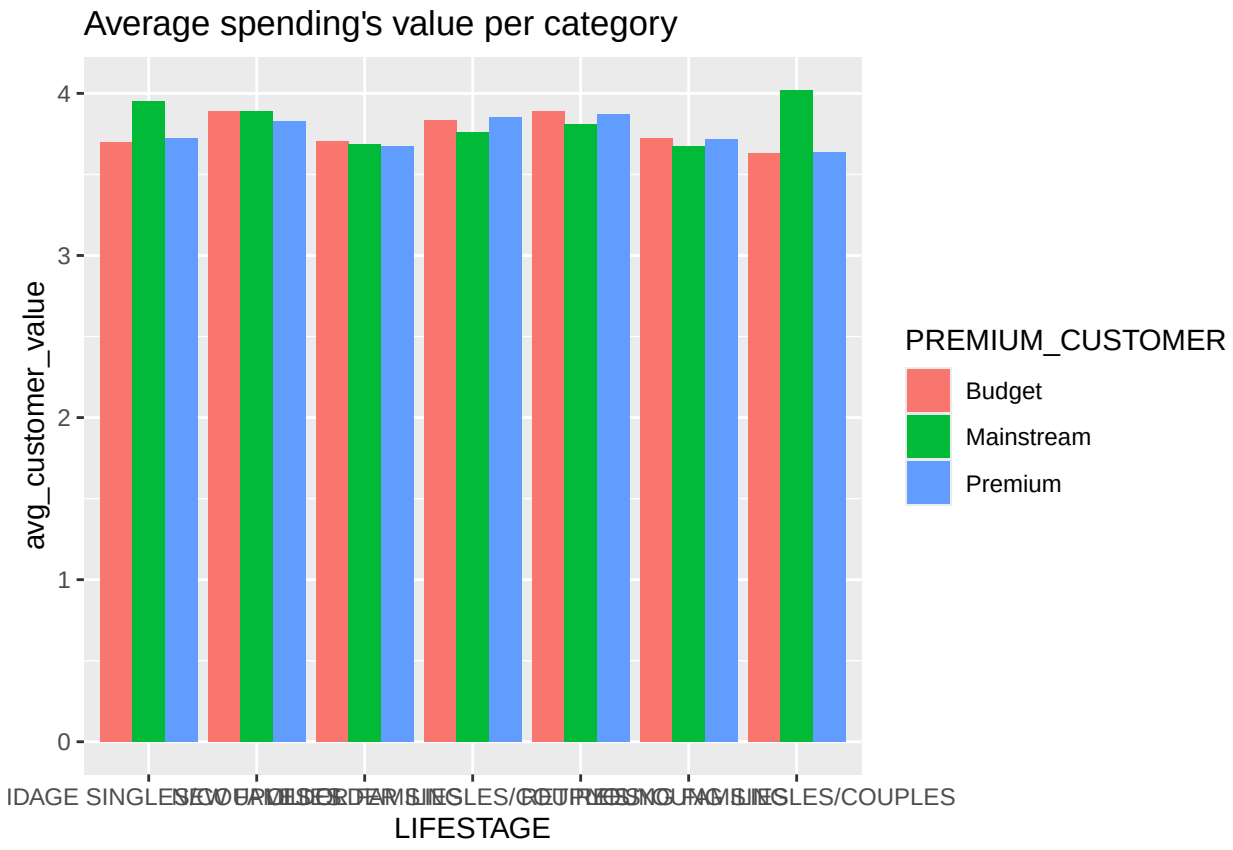

```
theme(axis.title.x = element_text(angle = 90, vjust = 0.5))
```

```
## List of 1
##  $ axis.title.x:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0.5
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
```

```
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

According to Average's size plot Mainstream/ Old families are likely to buy more chips globaly.

```
# how many do they spend per product (quality)?
avg_customer_value <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise( avg_customer_value = sum(TOT_SALES)/sum(PROD_QTY), .groups = "drop")

ggplot(avg_customer_value, aes(weight = avg_customer_value, x = LIFESTAGE,
                               fill = PREMIUM_CUSTOMER))+
  geom_bar(position = position_dodge())+
  labs (x = "LIFESTAGE", y = "avg_customer_value", title = "Average spending's value per category " )
```




```
theme(axis.title.x = element_text(angle = 90, vjust = 0.5))
```

```
## List of 1
##  $ axis.title.x:List of 11
##   ..$ family       : NULL
##   ..$ face         : NULL
##   ..$ colour       : NULL
##   ..$ size         : NULL
##   ..$ hjust        : NULL
##   ..$ vjust        : num 0.5
```

```
##   ..$ angle        : num 90
##   ..$ lineheight   : NULL
##   ..$ margin       : NULL
##   ..$ debug        : NULL
##   ..$ inherit.blank: logi FALSE
##   ..- attr(*, "class")= chr [1:2] "element_text" "element"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

Young single couple in mainstream are likely to spend more.The number of chips is not high because they want quality more than quantity. as the store interest is to know who spend more on chips, we can deepen the relationship until to show which brand they preferred. With t-test, we are going to compare the average spending amount between young single couple- mainstream and young single couple- others

```r
# Create the PRICE column
data <- data %>% mutate(PRICE = TOT_SALES / PROD_QTY)

# Subset: people from both groups
young_single_couple <- data %>%
  filter(LIFESTAGE =="YOUNG SINGLES/COUPLES")

# Group 1: Mainstream
group_mainstream <- young_single_couple %>%
  filter(PREMIUM_CUSTOMER == "Mainstream") %>%
  pull(PRICE)

# Group 2: Not Mainstream
group_non_mainstream <- young_single_couple %>%
  filter(PREMIUM_CUSTOMER != "Mainstream") %>%
  pull(PRICE)

# Run the t-test using 2 numeric vectors
t_test_result <- t.test(group_mainstream, group_non_mainstream, alternative = "greater")

# Print result
print(t_test_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  group_mainstream and group_non_mainstream
## t = 30.838, df = 22068, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3921898       Inf
## sample estimates:
## mean of x mean of y
##  4.021682  3.607394
```

Null hypothesis is rejected (p-value $<$ 2.2e-16) and in young_single_couple mainstream and non_mainstream are significantly differences in means.

```r
# let's sum up the behavior of customers

segments <- data %>%
  group_by(LIFESTAGE,PREMIUM_CUSTOMER) %>%
  summarise( revennue = sum(TOT_SALES),
             avg_spend = mean(PRICE),
             avg_pack_size = mean(PROD_SIZE),
             count = uniqueN(LYLTY_CARD_NBR), .groups = "drop")

print (segments)
```

```
## # A tibble: 21 x 6
##    LIFESTAGE             PREMIUM_CUSTOMER revennue avg_spend avg_pack_size count
##    <chr>                 <chr>               <dbl>     <dbl>          <dbl> <int>
##  1 MIDAGE SINGLES/COUPL~ Budget              24685       3.69           180.  1352
##  2 MIDAGE SINGLES/COUPL~ Mainstream          62412.      3.96           186.  3048
##  3 MIDAGE SINGLES/COUPL~ Premium             40146.      3.72           182.  2129
##  4 NEW FAMILIES          Budget              15349.      3.88           182.   970
##  5 NEW FAMILIES          Mainstream          11593.      3.88           183.   737
##  6 NEW FAMILIES          Premium              7978.      3.82           183.   513
##  7 OLDER FAMILIES        Budget             115917.      3.70           182.  4325
##  8 OLDER FAMILIES        Mainstream          70534.      3.69           182.  2628
##  9 OLDER FAMILIES        Premium             55477.      3.67           182.  2093
## 10 OLDER SINGLES/COUPLES Budget              92790       3.83           183.  4484
## # i 11 more rows
```

```r
test <- cor.test(data$PRICE,data$PROD_SIZE, method = "pearson")
print(test)
```

```
##
##  Pearson's product-moment correlation
##
## data:  data$PRICE and data$PROD_SIZE
## t = 176.62, df = 184681, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3762290 0.3840324
## sample estimates:
##       cor
## 0.3801375
```

```r
# BRAND affinity

group_mainstream_shr1 <- young_single_couple %>%
  filter(PREMIUM_CUSTOMER == "Mainstream") %>%
  group_by(BRANDS)%>%
  summarise(sales = sum(TOT_SALES)) %>%
  mutate(share_target_one = sales/sum(sales)*100)

group_mainstream_shr2 <- young_single_couple %>%
  filter(PREMIUM_CUSTOMER != "Mainstream") %>%
  group_by(BRANDS)%>%
```

```
  summarise(sales = sum(TOT_SALES)) %>%
  mutate(share_target_two = sales/sum(sales)*100)

#Affinty Score
 brand_affinity <- group_mainstream_shr1 %>%
   full_join(group_mainstream_shr2, by = "BRANDS") %>%
   mutate(affinity_ratio = share_target_one/share_target_two)
  print(brand_affinity)
```

```
## # A tibble: 22 x 6
##    BRANDS    sales.x share_target_one sales.y share_target_two affinity_ratio
##    <chr>       <dbl>            <dbl>   <dbl>            <dbl>          <dbl>
##  1 CCS          850.            0.786   1363.            1.93           0.407
##  2 CHEEZELS    3318.            3.07    1999.            2.83           1.08
##  3 COBS        6129             5.66    3690.            5.22           1.08
##  4 DORITOS    19396.           17.9    10288.           14.6            1.23
##  5 FRENCH       429             0.396    588             0.832          0.476
##  6 GRAIN       3791             3.50    2228.            3.15           1.11
##  7 GRNWVES      395.            0.365    496             0.702          0.520
##  8 INFUZIONS   6350.            5.87    3306.            4.68           1.25
##  9 KETTLE     20519            19.0    10443.           14.8            1.28
## 10 NATURAL     1734             1.60    2326.            3.29           0.487
## # i 12 more rows
```

Conclusion Young singles and couples in mainstream are likely to buy more chips and the preferred chips is the “TYRRELLS”. There is not proof of any relationship between the “PRICE” and “PROD_SIZE”. However, 190 g is preferred.

“‘