Retail Strategy & Customer Analytics - Transaction Data & Customer Analytcs

Kashif Ali

2022-08-12

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com (http://rmarkdown.rstudio.com).

```
#Load Required Libraries
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.1.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.3
```

```
library(ggmosaic)
```

```
## Warning: package 'ggmosaic' was built under R version 4.1.3
```

```
library(readr)
```

```
#Load the data
transdata <- read.csv("C:/Users/kashi/Downloads/QVI_transaction_data.csv")
purchasebehavior <- read.csv("C:/Users/kashi/Downloads/QVI_purchase_behaviour.csv")
```

##RUN EXPLORATORY ANALYSIS The first step in any analysis is to first understand the data.

```
#Examine Transaction Data
str(transdata)
```

```
## 'data.frame':    264836 obs. of  8 variables:
##  $ DATE          : chr  "2018-10-17" "2019-05-14" "2019-05-20" "2018-09-05" ...
##  $ STORE_NBR     : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ LYLTY_CARD_NBR: int  1000 1307 1343 1052 1081 1081 1081 1081 1184 1307 ...
##  $ TXN_ID        : int  1 348 383 57 92 93 94 95 216 346 ...
##  $ PROD_NBR      : int  5 66 61 44 17 96 8 57 2 96 ...
##  $ PROD_NAME     : chr  "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g"
"Smiths Crinkle Cut  Chips Chicken 170g" "Thins Chips Light&  Tangy 175g" ...
##  $ PROD_QTY      : int  2 3 2 1 1 2 1 1 1 2 ...
##  $ TOT_SALES     : num  6 6.3 2.9 3.3 4.6 3.8 2.9 5.1 3.8 3.8 ...
```

##Convert date to date format

```
transdata$DATE <- as.Date(transdata$DATE)
View(transdata)
```

```
#Examine PROD_NAME
setDT(transdata)
transdata[, .N , PROD_NAME]
```

```
##                                       PROD_NAME    N
##   1:   Natural Chip        Compny SeaSalt175g 1468
##   2:                 CCs Nacho Cheese    175g 1498
##   3:   Smiths Crinkle Cut  Chips Chicken 170g 1484
##   4:           Thins Chips Light&  Tangy 175g 3188
##   5:       Kettle Sensations   BBQ&Maple 150g 3083
##  ---
## 110:      Smiths Crinkle Cut  Snag&Sauce 150g 1503
## 111:   Thins Chips         Originl saltd 175g 1441
## 112: Cobs Popd Swt/Chlli &Sr/Cream Chips 110g 3269
## 113:                 Pringles Slt Vingar 134g 3095
## 114:          Pringles Original   Crisps 134g 3157
```

==Some Text Analysis==

```
#Check for any incorrect entries
prodWords <- data.table(unlist(strsplit(unique(transdata[, PROD_NAME]), " ")))
setnames(prodWords, 'words')
```

Remove all words with digits and special characters

```
#remove digits
prodWords <- prodWords[grepl("\\d", words) == FALSE, ]
#remove special characters
prodWords <- prodWords[grepl("[:alpha:]" , words), ]
```

```
#count frequency of words & sort by highest to lowest
prodWords[, .N, words][order(N, decreasing = TRUE)]
```

```
##          words  N
##   1:     Chips 21
##   2:    Smiths 16
##   3:   Crinkle 14
##   4:    Kettle 13
##   5:    Cheese 12
##  ---
## 127:     saltd  1
## 128: Swt/Chlli  1
## 129: &Sr/Cream  1
## 130:       Slt  1
## 131:    Vingar  1
```

As we are only interested in chips, so we will remove salsa products.

```
#remove salsa products
transdata[, SALSA := grepl("salsa", tolower(PROD_NAME))]
transdata <- transdata[SALSA == FALSE, ][, SALSA := NULL]
```

==Summary of Data to Check Nulls and Outliers

```
summary(transdata)
```

```
##       DATE              STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :2018-07-01   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  70015   1st Qu.:  67569
##  Median :2018-12-30   Median :130.0   Median : 130367   Median : 135183
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135531   Mean   : 135131
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203084   3rd Qu.: 202654
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME           PROD_QTY         TOT_SALES
##  Min.   :  1.00   Length:246742      Min.   :  1.000   Min.   :  1.700
##  1st Qu.: 26.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.800
##  Median : 53.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.35                      Mean   :  1.908   Mean   :  7.321
##  3rd Qu.: 87.00                      3rd Qu.:  2.000   3rd Qu.:  8.800
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
```

```
#Filter data set to find outliers
transdata[PROD_QTY == 200, ]
```

```
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 1: 2018-08-19       226         226000 226201        4
## 2: 2019-05-20       226         226000 226210        4
##                           PROD_NAME PROD_QTY TOT_SALES
## 1: Dorito Corn Chp     Supreme 380g      200       650
## 2: Dorito Corn Chp     Supreme 380g      200       650
```

```
#Let's see if customer has some other transactions
transdata[LYLTY_CARD_NBR == 226000, ]
```

```
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 1: 2018-08-19       226         226000 226201        4
## 2: 2019-05-20       226         226000 226210        4
##                            PROD_NAME PROD_QTY TOT_SALES
## 1: Dorito Corn Chp     Supreme 380g      200       650
## 2: Dorito Corn Chp     Supreme 380g      200       650
```

```
#Filter out the customer based on loyalty card number
transdata <- transdata[LYLTY_CARD_NBR != 226000, ]
```

```
#Summary of data
summary(transdata)
```

```
##       DATE              STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :2018-07-01   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  70015   1st Qu.:  67569
##  Median :2018-12-30   Median :130.0   Median : 130367   Median : 135182
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135530   Mean   : 135130
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203083   3rd Qu.: 202652
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:246740      Min.   :1.000   Min.   : 1.700
##  1st Qu.: 26.00   Class :character   1st Qu.:2.000   1st Qu.: 5.800
##  Median : 53.00   Mode  :character   Median :2.000   Median : 7.400
##  Mean   : 56.35                      Mean   :1.906   Mean   : 7.316
##  3rd Qu.: 87.00                      3rd Qu.:2.000   3rd Qu.: 8.800
##  Max.   :114.00                      Max.   :5.000   Max.   :29.500
```

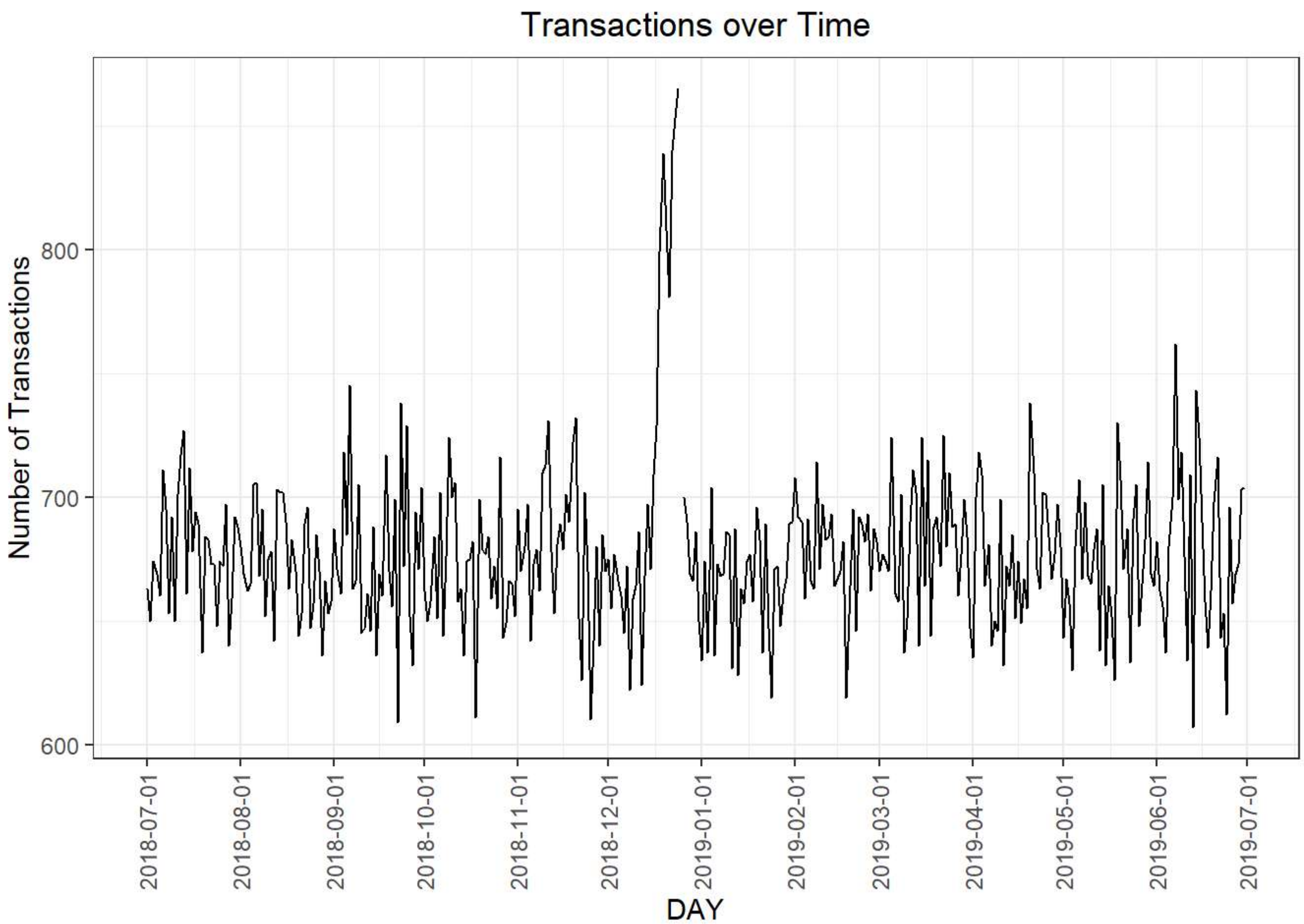
```
#count the number of transaction by date
transdata[, .N, by= DATE]
```

```
##            DATE   N
##   1: 2018-10-17 682
##   2: 2019-05-14 705
##   3: 2019-05-20 707
##   4: 2018-09-05 685
##   5: 2018-09-27 632
##  ---
## 360: 2018-08-19 670
## 361: 2018-11-15 689
## 362: 2019-04-15 651
## 363: 2019-01-22 689
## 364: 2019-05-03 657
```

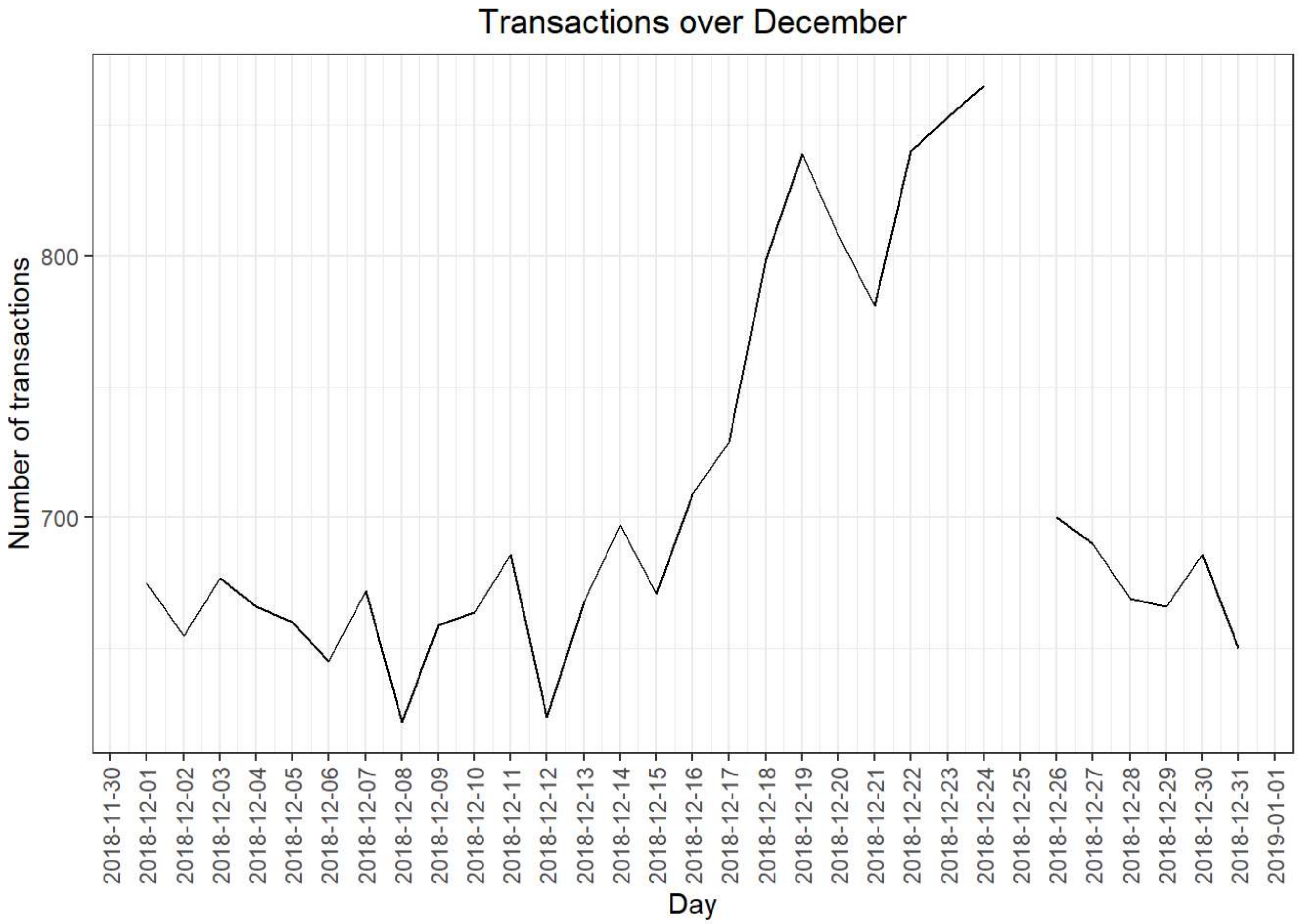
```
#create a sequence of dates
eachdate <- data.table(seq(as.Date("2018/07/01"), as.Date("2019/06/30"), by = "day"))
setnames(eachdate, "DATE")
byday_trans <- merge(eachdate, transdata[, .N, by = DATE], all.x = TRUE)
```

```
#SETTING PLOT THEMES TO FORMAT GRAPHS
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
```

```
# over time transactions plot
ggplot(byday_trans, 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))
```




```
#narrow down number of days
ggplot(byday_trans[month(DATE) == 12, ], aes(x = DATE, y = N)) +
geom_line() +
labs(x = "Day", y = "Number of transactions", title = "Transactions over December") +
scale_x_date(breaks = "1 day") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



This graph shows that higher sales were in December because arrival of christmas. It is now confirmed that no more outliers are present in the transaction data.

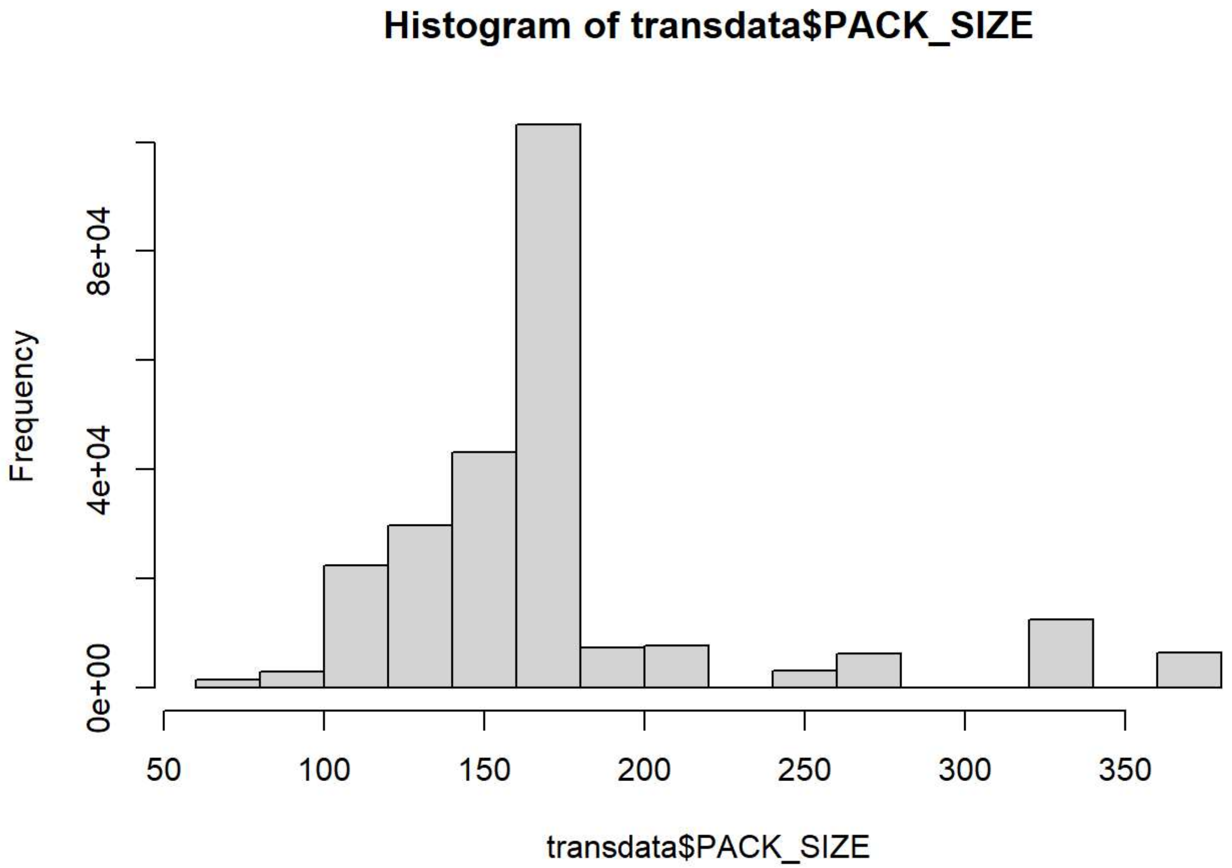
```
#pack size
transdata[, PACK_SIZE := parse_number(PROD_NAME)]

transdata[, .N, PACK_SIZE][order(PACK_SIZE)]
```

```
##     PACK_SIZE     N
##  1:        70  1507
##  2:        90  3008
##  3:       110 22387
##  4:       125  1454
##  5:       134 25102
##  6:       135  3257
##  7:       150 40203
##  8:       160  2970
##  9:       165 15297
## 10:       170 19983
## 11:       175 66390
## 12:       180  1468
## 13:       190  2995
## 14:       200  4473
## 15:       210  6272
## 16:       220  1564
## 17:       250  3169
## 18:       270  6285
## 19:       330 12540
## 20:       380  6416
```

```
transdata
```

```
##               DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##      1: 2018-10-17         1           1000      1        5
##      2: 2019-05-14         1           1307    348       66
##      3: 2019-05-20         1           1343    383       61
##      4: 2018-09-05         1           1052     57       44
##      5: 2018-09-27         1           1081     92       17
##     ---
## 246736: 2019-03-09       272         272319 270088       89
## 246737: 2018-08-13       272         272358 270154       74
## 246738: 2018-11-06       272         272379 270187       51
## 246739: 2018-12-27       272         272379 270188       42
## 246740: 2018-09-22       272         272380 270189       74
##                                  PROD_NAME PROD_QTY TOT_SALES PACK_SIZE
##      1:  Natural Chip        Compny SeaSalt175g        2       6.0       175
##      2:                CCs Nacho Cheese    175g        3       6.3       175
##      3:  Smiths Crinkle Cut  Chips Chicken 170g        2       2.9       170
##      4:          Thins Chips Light&  Tangy 175g        1       3.3       175
##      5:      Kettle Sensations   BBQ&Maple 150g        1       4.6       150
##     ---
## 246736: Kettle Sweet Chilli And Sour Cream 175g        2      10.8       175
## 246737:           Tostitos Splash Of  Lime 175g        1       4.4       175
## 246738:                Doritos Mexicana    170g        2       8.8       170
## 246739: Doritos Corn Chip Mexican Jalapeno 150g        2       7.8       150
## 246740:           Tostitos Splash Of  Lime 175g        2       8.8       175
```

```
#let's plot histogram
hist (transdata$PACK_SIZE)
```




```
#now we will create BRANDS & for this purpose we can use first word in PROD_NAME
transdata[, BRAND := toupper(substr(PROD_NAME, 1, regexpr(pattern = ' ', PROD_NAME) -1))]

transdata[, .N, by = BRAND][order(-N)]
```

```
##          BRAND     N
##  1:     KETTLE 41288
##  2:     SMITHS 27390
##  3:   PRINGLES 25102
##  4:    DORITOS 22041
##  5:      THINS 14075
##  6:        RRD 11894
##  7:  INFUZIONS 11057
##  8:         WW 10320
##  9:       COBS  9693
## 10:   TOSTITOS  9471
## 11:   TWISTIES  9454
## 12:   TYRRELLS  6442
## 13:      GRAIN  6272
## 14:    NATURAL  6050
## 15:   CHEEZELS  4603
## 16:        CCS  4551
## 17:        RED  4427
## 18:     DORITO  3183
## 19:     INFZNS  3144
## 20:      SMITH  2963
## 21:    CHEETOS  2927
## 22:      SNBTS  1576
## 23:     BURGER  1564
## 24: WOOLWORTHS  1516
## 25:    GRNWVES  1468
## 26:   SUNBITES  1432
## 27:        NCC  1419
## 28:     FRENCH  1418
##          BRAND     N
```

```
#combine same brand names
transdata[BRAND == "RED", BRAND := "RRD"]
transdata[BRAND == "SNBTS", BRAND := "SUNBITES"]
transdata[BRAND == "INFZNS", BRAND := "INFUZIONS"]
transdata[BRAND == "WW", BRAND := "WOOLWORTHS"]
transdata[BRAND == "SMITH", BRAND := "SMITHS"]
transdata[BRAND == "NCC", BRAND := "NATURAL"]
transdata[BRAND == "DORITO", BRAND := "DORITOS"]
transdata[BRAND == "GRAIN", BRAND := "GRNWVES"]
```

```
#again check
transdata[, .N, by = BRAND][order(N)]
```

```
##          BRAND     N
##  1:     FRENCH  1418
##  2:     BURGER  1564
##  3:    CHEETOS  2927
##  4:   SUNBITES  3008
##  5:        CCS  4551
##  6:   CHEEZELS  4603
##  7:   TYRRELLS  6442
##  8:    NATURAL  7469
##  9:    GRNWVES  7740
## 10:   TWISTIES  9454
## 11:   TOSTITOS  9471
## 12:       COBS  9693
## 13: WOOLWORTHS 11836
## 14:      THINS 14075
## 15:  INFUZIONS 14201
## 16:        RRD 16321
## 17:   PRINGLES 25102
## 18:    DORITOS 25224
## 19:     SMITHS 30353
## 20:     KETTLE 41288
```

Examining Customer Data

```
str(purchasebehavior)
```

```
## 'data.frame':    72637 obs. of  3 variables:
##  $ LYLTY_CARD_NBR  : int  1000 1002 1003 1004 1005 1007 1009 1010 1011 1012 ...
##  $ LIFESTAGE       : chr  "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "O
LDER SINGLES/COUPLES" ...
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream" ...
```

```
summary(purchasebehavior)
```

```
##  LYLTY_CARD_NBR     LIFESTAGE         PREMIUM_CUSTOMER
##  Min.   :   1000   Length:72637       Length:72637
##  1st Qu.:  66202   Class :character   Class :character
##  Median : 134040   Mode  :character   Mode  :character
##  Mean   : 136186
##  3rd Qu.: 203375
##  Max.   :2373711
```

```
#Examine Values of Lifestage & Premium Customer
setDT(purchasebehavior)
purchasebehavior[,.N, by= LIFESTAGE][order(-N)]
```

```
##                  LIFESTAGE     N
## 1:               RETIREES 14805
## 2:  OLDER SINGLES/COUPLES 14609
## 3:  YOUNG SINGLES/COUPLES 14441
## 4:         OLDER FAMILIES  9780
## 5:         YOUNG FAMILIES  9178
## 6: MIDAGE SINGLES/COUPLES  7275
## 7:           NEW FAMILIES  2549
```

```
purchasebehavior[, .N, by= PREMIUM_CUSTOMER][order(-N)]
```

```
##    PREMIUM_CUSTOMER     N
## 1:       Mainstream 29245
## 2:           Budget 24470
## 3:          Premium 18922
```

Now merge transdata and purchasebehaviour data

```
mergdata <- merge(transdata, purchasebehavior, all.x = TRUE)
```

```
#Let’s also check if some customers were not matched on by checking for nulls.
mergdata[is.null(LIFESTAGE), .N]
```

```
## [1] 0
```

```
mergdata[is.null(PREMIUM_CUSTOMER), .N]
```

```
## [1] 0
```

BRAVO! DATA EXPLORATION IS COMPLETED.

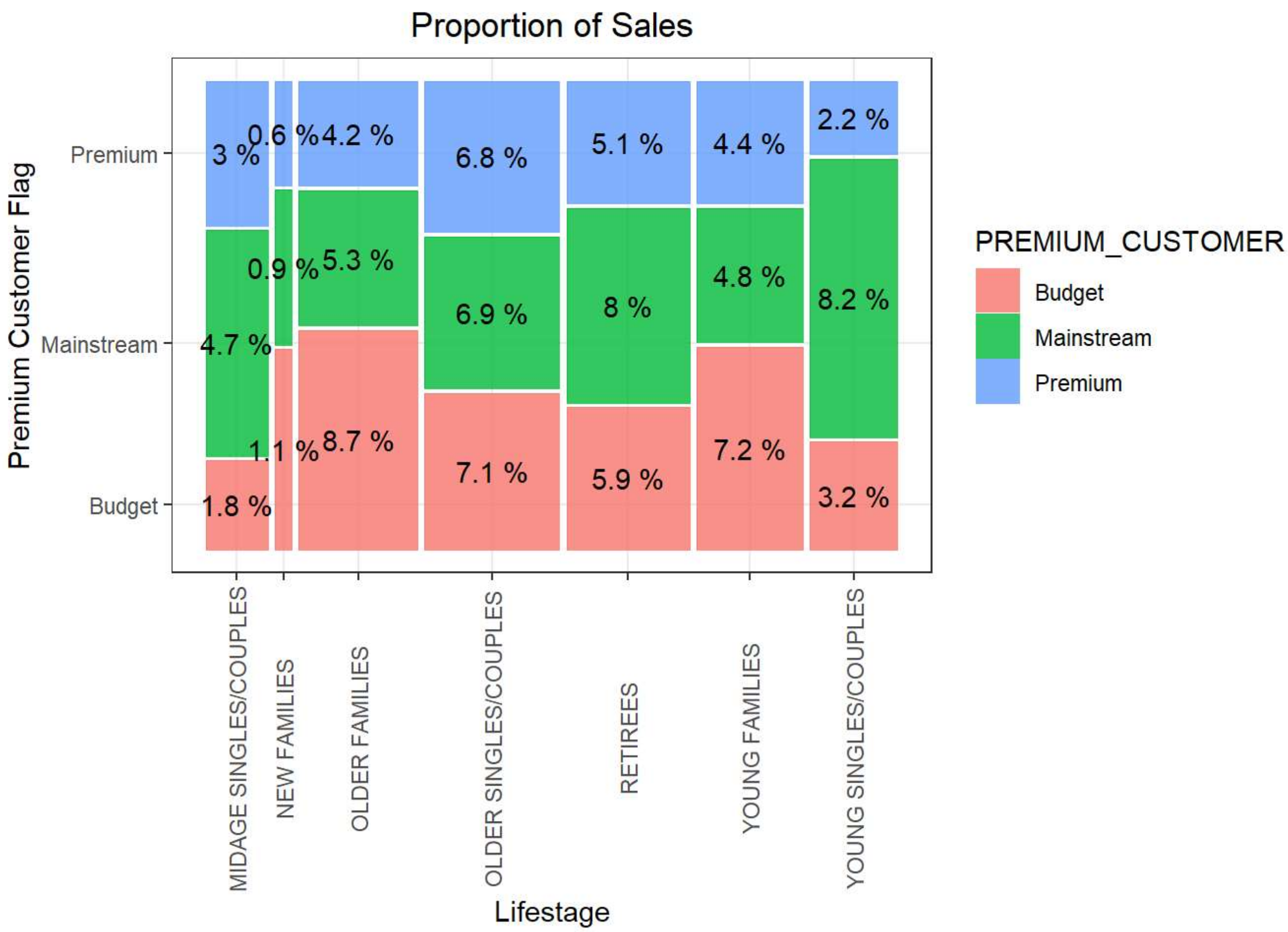
Data Analysis on Customer Segmentation

```
##Total sales by LIFESTAGE & PREMIUM_CUSTOMERS
setDT(mergdata)
sales <- mergdata[, .(SALES = sum(TOT_SALES)), .(LIFESTAGE, PREMIUM_CUSTOMER)]
```

```
#Create a plot
plt <- ggplot(data = sales) +
  geom_mosaic(aes(weight = SALES, x= product(PREMIUM_CUSTOMER, LIFESTAGE), fill = PREMIUM_CUSTOM
ER)) +
  labs(x= "Lifestage", y= "Premium Customer Flag", title = "Proportion of Sales") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

```
#Plot and Lable with Proportion of Sales
plt + geom_text(data = ggplot_build(plt)$data[[1]], aes(x = (xmin + xmax)/2 , y =
(ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)*100,
'%'))))
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## Please use `unite()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

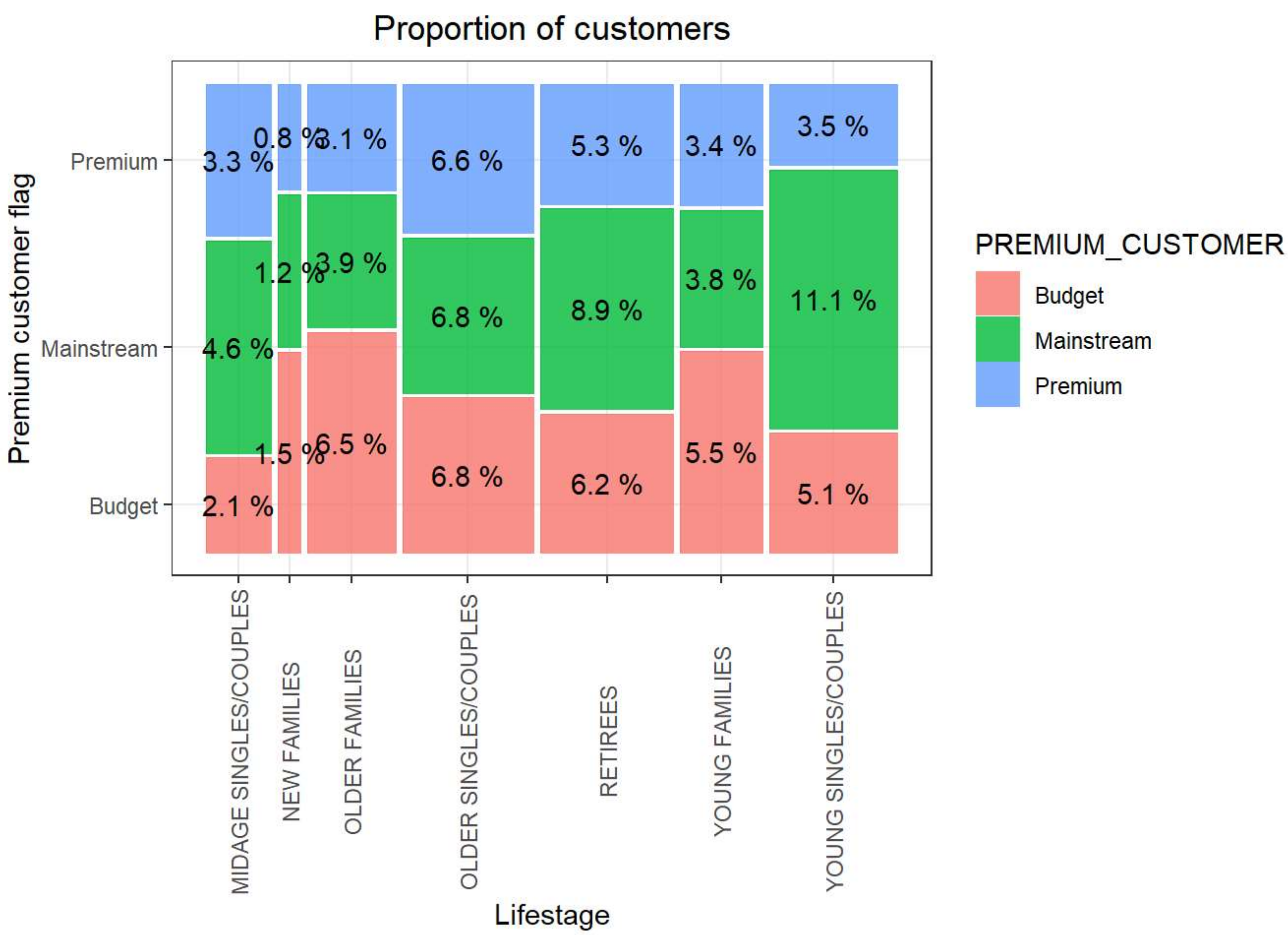


Let's see if the higher sales are due to there being more customers who buy chips.

```
### Number of customers by LIFESTAGE and PREMIUM_CUSTOMER
customers <- mergdata[, .(CUSTOMERS = uniqueN(LYLTY_CARD_NBR)), .(LIFESTAGE, PREMIUM_CUSTOMER)]
[order(-CUSTOMERS)]
```

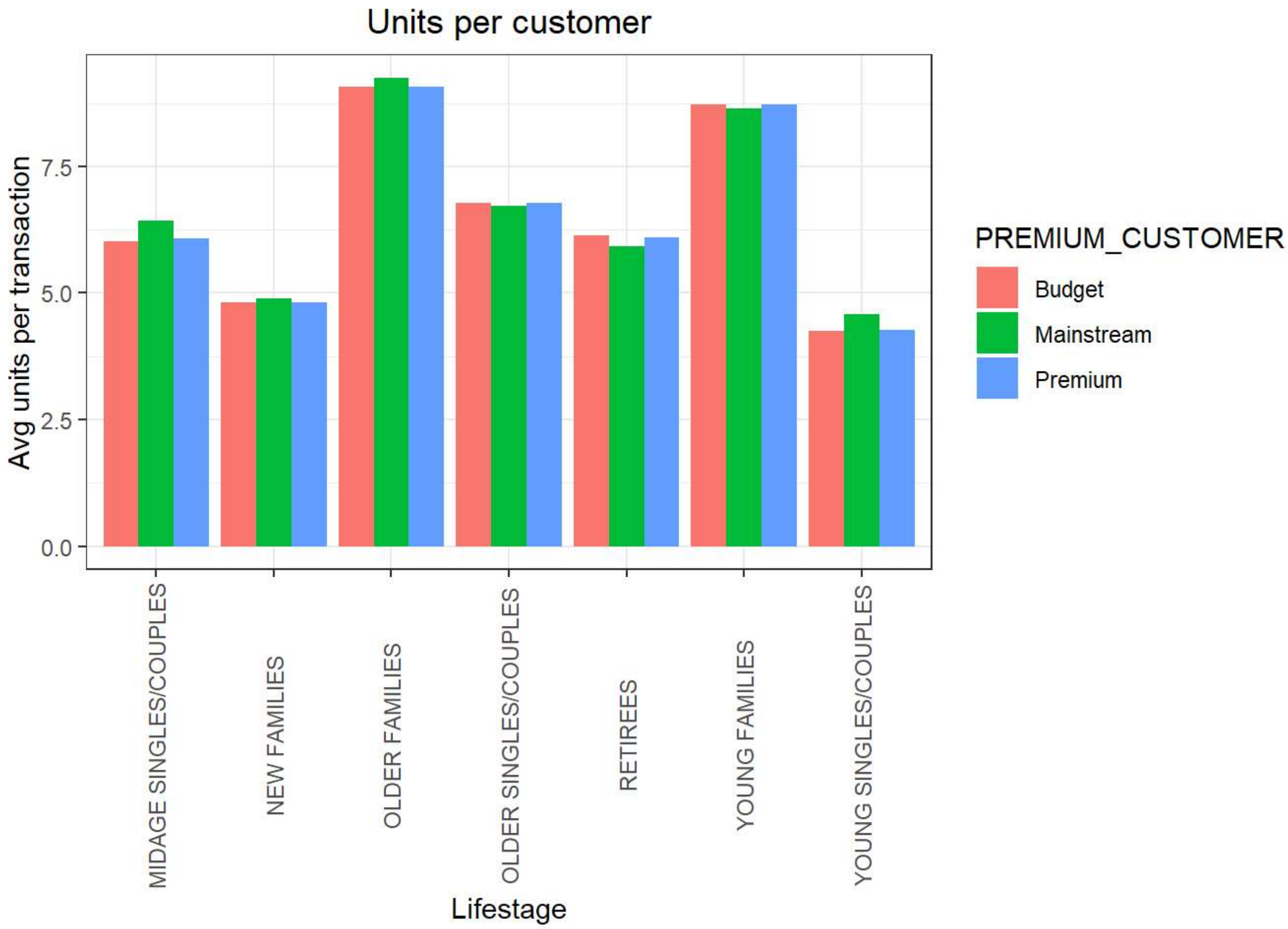
```
#create a plot
p <- ggplot(data = customers) +
geom_mosaic(aes(weight = CUSTOMERS, x = product(PREMIUM_CUSTOMER, LIFESTAGE), fill = PREMIUM_CUS
TOMER)) +
labs(x = "Lifestage", y = "Premium customer flag", title = "Proportion of customers") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

```
#plot and label with proportion of customers
p + geom_text(data = ggplot_build(p)$data[[1]], aes(x = (xmin + xmax)/2 , y =
(ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)*100,
'%'))))
```



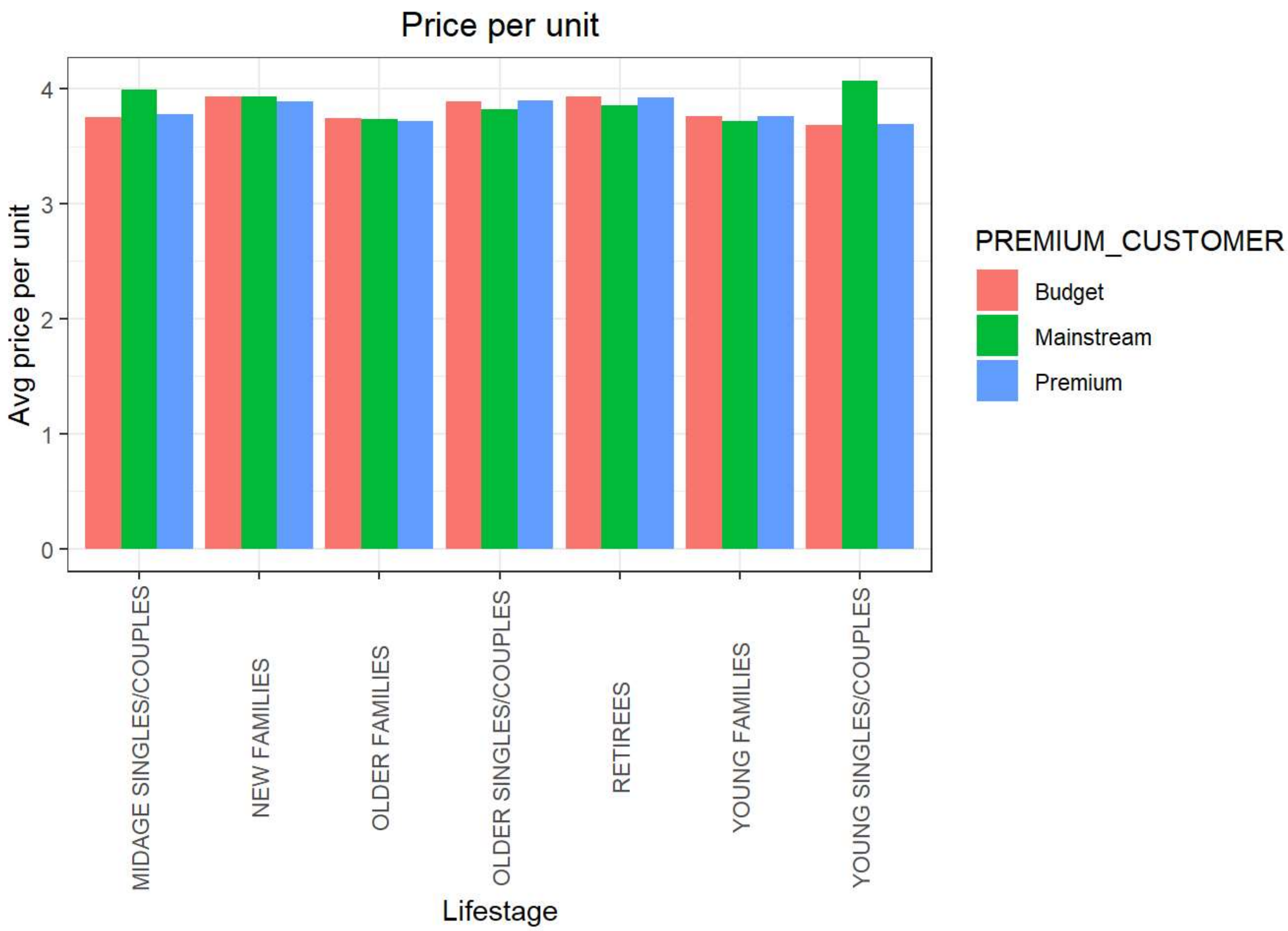

```
# Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER
avg_units <- mergdata[, .(AVG = sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR)), .(LIFESTAGE, PREMIUM_CUS
TOMER)][order(-AVG)]
```

```
#### Create plot
ggplot(data = avg_units, aes(weight = AVG, x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) +
geom_bar(position = position_dodge()) +
labs(x = "Lifestage", y = "Avg units per transaction", title = "Units per customer") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



Let's also investigate the average price per unit chips bought for each customer segment as this is also a driver of total sales.

```
## Average price per unit by LIFESTAGE and PREMIUM_CUSTOMER
avg_price <- mergdata[, .(AVG = sum(TOT_SALES)/sum(PROD_QTY)), .(LIFESTAGE, PREMIUM_CUSTOMER)][o
rder(-AVG)]


# Create plot
ggplot(data = avg_price, aes(weight = AVG, x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) +
geom_bar(position = position_dodge()) +
labs(x = "Lifestage", y = "Avg price per unit", title = "Price per unit") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



As the difference in average price per unit isn't large, we can check if this difference is statistically different.

```
#### Perform an independent t-test between mainstream vs premium and budget midage and

#### young singles and couples
pricePerUnit <- mergdata[, price := TOT_SALES/PROD_QTY]
t.test(mergdata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CU
STOMER == "Mainstream", price]
, mergdata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOME
R != "Mainstream", price]
, alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  mergdata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM
_CUSTOMER == "Mainstream", price] and mergdata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE
SINGLES/COUPLES") & PREMIUM_CUSTOMER != "Mainstream", price]
## t = 37.624, df = 54791, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3187234       Inf
## sample estimates:
## mean of x mean of y
##  4.039786  3.706491
```

The t-test results in a p-value < 2.2e-16, i.e. the unit price for mainstream, young and mid-age singles and couples are significantly higher than that of budget or premium, young and midage singles and couples.

Deep dive into specific customer segments for insights

```
#### Deep dive into Mainstream, young singles/couples
segment1 <- mergdata[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER == "Mainstream",]
other <- mergdata[!(LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER == "Mainstream"),]

#### Brand affinity compared to the rest of the population
quantity_segment1 <- segment1[, sum(PROD_QTY)]

quantity_other <- other[, sum(PROD_QTY)]
quantity_segment1_by_brand <- segment1[, .(targetSegment = sum(PROD_QTY)/quantity_segment1), by
 = BRAND]
quantity_other_by_brand <- other[, .(other = sum(PROD_QTY)/quantity_other), by = BRAND]
brand_proportions <- merge(quantity_segment1_by_brand, quantity_other_by_brand)[, affinityToBran
d := targetSegment/other]
brand_proportions[order(-affinityToBrand)]
```

```
##          BRAND targetSegment       other affinityToBrand
##  1:   TYRRELLS   0.031552795 0.025692464       1.2280953
##  2:   TWISTIES   0.046183575 0.037876520       1.2193194
##  3:    DORITOS   0.122760524 0.101074684       1.2145526
##  4:     KETTLE   0.197984817 0.165553442       1.1958967
##  5:   TOSTITOS   0.045410628 0.037977861       1.1957131
##  6:   PRINGLES   0.119420290 0.100634769       1.1866703
##  7:       COBS   0.044637681 0.039048861       1.1431238
##  8:  INFUZIONS   0.064679089 0.057064679       1.1334347
##  9:      THINS   0.060372671 0.056986370       1.0594230
## 10:    GRNWVES   0.032712215 0.031187957       1.0488733
## 11:   CHEEZELS   0.017971014 0.018646902       0.9637534
## 12:     SMITHS   0.096369910 0.124583692       0.7735355
## 13:     FRENCH   0.003947550 0.005758060       0.6855694
## 14:    CHEETOS   0.008033126 0.012066591       0.6657329
## 15:        RRD   0.043809524 0.067493678       0.6490908
## 16:    NATURAL   0.019599724 0.030853989       0.6352412
## 17:        CCS   0.011180124 0.018895650       0.5916771
## 18:   SUNBITES   0.006349206 0.012580210       0.5046980
## 19: WOOLWORTHS   0.024099379 0.049427188       0.4875733
## 20:     BURGER   0.002926156 0.006596434       0.4435967
```

We can see that : • Mainstream young singles/couples are 23% more likely to purchase Tyrrells chips compared to the rest of the population • Mainstream young singles/couples are 56% less likely to purchase Burger Rings compared to the rest of the population

Let's also find out if our target segment tends to buy larger packs of chips.

```
### Preferred pack size compared to the rest of the population
quantity_segment1_by_pack <- segment1[, .(targetSegment = sum(PROD_QTY)/quantity_segment1), by =
PACK_SIZE]

quantity_other_by_pack <- other[, .(other = sum(PROD_QTY)/quantity_other), by = PACK_SIZE]

pack_proportions <- merge(quantity_segment1_by_pack, quantity_other_by_pack)[, affinityToPack :=
targetSegment/other]

pack_proportions[order(-affinityToPack)]
```

```
##     PACK_SIZE targetSegment       other affinityToPack
##  1:       270   0.031828847 0.025095929      1.2682873
##  2:       380   0.032160110 0.025584213      1.2570295
##  3:       330   0.061283644 0.050161917      1.2217166
##  4:       134   0.119420290 0.100634769      1.1866703
##  5:       110   0.106280193 0.089791190      1.1836372
##  6:       210   0.029123533 0.025121265      1.1593180
##  7:       135   0.014768806 0.013075403      1.1295106
##  8:       250   0.014354727 0.012780590      1.1231662
##  9:       170   0.080772947 0.080985964      0.9973697
## 10:       150   0.157598344 0.163420656      0.9643722
## 11:       175   0.254989648 0.270006956      0.9443818
## 12:       165   0.055652174 0.062267662      0.8937572
## 13:       190   0.007481021 0.012442016      0.6012708
## 14:       180   0.003588682 0.006066692      0.5915385
## 15:       160   0.006404417 0.012372920      0.5176157
## 16:        90   0.006349206 0.012580210      0.5046980
## 17:       125   0.003008972 0.006036750      0.4984423
## 18:       200   0.008971705 0.018656115      0.4808989
## 19:        70   0.003036577 0.006322350      0.4802924
## 20:       220   0.002926156 0.006596434      0.4435967
```

It looks like Mainstream young singles/couples are 27% more likely to purchase a 270g pack of chips compared to the rest of the population but let’s dive into what brands sell this pack size.

```
mergdata[PACK_SIZE == 270, unique(PROD_NAME)]
```

```
## [1] "Twisties Cheese     270g" "Twisties Chicken270g"
```

Twisties are the only brand offering 270g packs and so this may instead be reflecting a higher likelihood of purchasing Twisties.

Conclusion Let’s recap what we’ve found! Sales have mainly been due to Budget - older families, Mainstream - young singles/couples, and Mainstream - retirees shoppers. We found that the high spend in chips for mainstream young singles/couples and retirees is due to there being more of them than other buyers. Mainstream, midage and young singles and couples are also more likely to pay more per packet of chips. This is indicative of impulse buying behaviour. We’ve also found that Mainstream young singles and couples are 23% more likely to purchase Tyrrells chips compared to the rest of the population. The Category Manager may want to increase the category’s performance by off-locating some Tyrrells and smaller packs of chips in discretionary space near segments where young singles and couples frequent more often to increase visibilty and impulse behaviour.

I can help the Category Manager with recommendations of where these segments are and further help them with measuring the impact of the changed placement. We’ll work on measuring the impact of trials in the next task and putting all these together in the third task.