Quantium DA Task1

Santosh Reddy Edulapalle

2023-02-13

```
library(readxl)
library(data.table)
library(ggplot2)
library(ggmosaic)
library(readr)
library(stringr)
library(dplyr)
```

Load required libraries

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:data.table':
##
##     between, first, last
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(arules)
```

```
## Loading required package: Matrix
##
## Attaching package: 'arules'
## The following object is masked from 'package:dplyr':
##
##     recode
## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```
library(arulesViz)
```

```
trans <- read_excel('QVI_transaction_data.xlsx')
```

Loading the dataset

Exploratory data analysis on Transaction data

The first step in any analysis is to first understand the data. Let's take a look at each of the datasets provided.

```
#showing head ( top 10 rows)
head(trans)
```

Examine transaction data

```
## # A tibble: 6 x 8
##    DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME       PROD_~1 TOT_S~2
##   <dbl>     <dbl>          <dbl>  <dbl>    <dbl> <chr>             <dbl>   <dbl>
## 1 43390         1           1000      1        5 Natural Chip  ~       2     6
## 2 43599         1           1307    348       66 CCs Nacho Chee~       3     6.3
## 3 43605         1           1343    383       61 Smiths Crinkle~       2     2.9
## 4 43329         2           2373    974       69 Smiths Chip Th~       5    15
## 5 43330         2           2426   1038      108 Kettle Tortill~       3    13.8
## 6 43604         4           4074   2982       57 Old El Paso Sa~       1     5.1
## # ... with abbreviated variable names 1: PROD_QTY, 2: TOT_SALES
```

```
#showing summary
summary(trans)
```

```
##       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   Min.   :  1.500
##  1st Qu.: 28.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.400
##  Median : 56.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.58                      Mean   :  1.907   Mean   :  7.304
##  3rd Qu.: 85.00                      3rd Qu.:  2.000   3rd Qu.:  9.200
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
```

```
#showing high level structure
str(trans)
```

```
## tibble [264,836 x 8] (S3: tbl_df/tbl/data.frame)
##  $ DATE          : num [1:264836] 43390 43599 43605 43329 43330 ...
##  $ STORE_NBR     : num [1:264836] 1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: num [1:264836] 1000 1307 1343 2373 2426 ...
##  $ TXN_ID        : num [1:264836] 1 348 383 974 1038 ...
##  $ PROD_NBR      : num [1:264836] 5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr [1:264836] "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g
##  $ PROD_QTY      : num [1:264836] 2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num [1:264836] 6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...
```

Convert DATE column to a date format We can see that the DATE type is DOUBLE We need to convert it to DATE type CSV and Excel integer dates begin on 30 Dec 1899

```
typeof(trans$DATE)
```

```
## [1] "double"
```

```
trans$DATE <- as.Date(trans$DATE,origin = '1899-12-30')
typeof(trans$DATE)
```

```
## [1] "double"
```

```
#examine structure
str(trans)
```

```
## tibble [264,836 x 8] (S3: tbl_df/tbl/data.frame)
##  $ DATE          : Date[1:264836], format: "2018-10-17" "2019-05-14" ...
##  $ STORE_NBR     : num [1:264836] 1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: num [1:264836] 1000 1307 1343 2373 2426 ...
##  $ TXN_ID        : num [1:264836] 1 348 383 974 1038 ...
##  $ PROD_NBR      : num [1:264836] 5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr [1:264836] "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g
##  $ PROD_QTY      : num [1:264836] 2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num [1:264836] 6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...
```

Examine PROD_NAME Since PROD_NAME is a name given to individual object, we will factorise it and make them into groups.

```
trans$PROD_NAME_FACTOR <- factor(trans$PROD_NAME)
summary(trans$PROD_NAME_FACTOR)
```

```
##   Kettle Mozzarella   Basil & Pesto 175g
##                                       3304
## Kettle Tortilla ChpsHny&Jlpno Chili 150g
##                                       3296
## Cobs Popd Swt/Chlli &Sr/Cream Chips 110g
##                                       3269
##   Tyrrells Crisps     Ched & Chives 165g
##                                       3268
##           Cobs Popd Sea Salt  Chips 110g
##                                       3265
##             Kettle 135g Swt Pot Sea Salt
##                                       3257
##            Tostitos Splash Of  Lime 175g
##                                       3252
## Infuzions Thai SweetChili PotatoMix 110g
##                                       3242
##   Smiths Crnkle Chip  Orgnl Big Bag 380g
##                                       3233
##     Thins Potato Chips  Hot & Spicy 175g
##                                       3229
## Kettle Sensations   Camembert & Fig 150g
##                                       3219
##  Doritos Corn Chips  Cheese Supreme 170g
##                                       3217
##                 Pringles Barbeque   134g
##                                       3210
##  Doritos Corn Chip Mexican Jalapeno 150g
##                                       3204
##  Kettle Sweet Chilli And Sour Cream 175g
##                                       3200
## Smiths Crinkle Chips Salt & Vinegar 330g
```

```
##                                     3197
##           Thins Chips Light&  Tangy 175g
##                                     3188
##         Dorito Corn Chp     Supreme 380g
##                                     3185
##             Pringles Sweet&Spcy BBQ 134g
##                                     3177
##  Infuzions BBQ Rib   Prawn Crackers 110g
##                                     3174
##  Tyrrells Crisps     Lightly Salted 165g
##                                     3174
##     Kettle Sea Salt     And Vinegar 175g
##                                     3173
##  Doritos Corn Chip Southern Chicken 150g
##                                     3172
##                     Twisties Chicken270g
##                                     3170
##          Twisties Cheese     Burger 250g
##                                     3169
##    Grain Waves         Sweet Chilli 210g
##                                     3167
##           Pringles SourCream  Onion 134g
##                                     3162
##    Doritos Corn Chips  Nacho Cheese 170g
##                                     3160
##   Cobs Popd Sour Crm  &Chives Chips 110g
##                                     3159
##                     Kettle Original 175g
##                                     3159
##          Pringles Original   Crisps 134g
##                                     3157
##                     Cheezels Cheese 330g
##                                     3149
##         Kettle Honey Soy    Chicken 175g
##                                     3148
##  Kettle Tortilla ChpsBtroot&Ricotta 150g
##                                     3146
##        Tostitos Smoked     Chipotle 175g
##                                     3145
##   Infzns Crn Crnchers Tangy Gcamole 110g
##                                     3144
##        Smiths Crinkle      Original 330g
##                                     3142
##     Kettle Tortilla ChpsFeta&Garlic 150g
##                                     3138
## Infuzions SourCream&Herbs Veg Strws 110g
##                                     3134
##    Kettle Sensations   Siracha Lime 150g
##                                     3127
## Old El Paso Salsa   Dip Chnky Tom Ht300g
##                                     3125
##        Doritos Corn Chips  Original 170g
##                                     3121
##                 Doritos Mexicana    170g
```

```
##                                     3115
##                 Twisties Cheese     270g
##                                     3115
##  Old El Paso Salsa   Dip Tomato Med 300g
##                                     3114
##         Pringles Mystery    Flavour 134g
##                                     3114
##         Thins Chips Seasonedchicken 175g
##                                     3114
##    Grain Waves Sour    Cream&Chives 210G
##                                     3105
##      Pringles Chicken    Salt Crips 134g
##                                     3104
##         Thins Chips Salt &  Vinegar 175g
##                                     3103
##                 Pringles Slt Vingar 134g
##                                     3095
## Old El Paso Salsa   Dip Tomato Mild 300g
##                                     3085
##       Kettle Sensations   BBQ&Maple 150g
##                                     3083
##         Pringles Sthrn FriedChicken 134g
##                                     3083
##          Tostitos Lightly    Salted 175g
##                                     3074
##         Doritos Cheese      Supreme 330g
##                                     3052
##                       Kettle Chilli 175g
##                                     3038
##    Smiths Chip Thinly  Cut Original 175g
##                                     1614
##     Snbts Whlgrn Crisps Cheddr&Mstrd 90g
##                                     1576
##  Natural Chip Co     Tmato Hrb&Spce 175g
##                                     1572
##                         Burger Rings 220g
##                                     1564
##   Natural ChipCo Sea  Salt & Vinegr 175g
##                                     1550
##                 CCs Tasty Cheese    175g
##                                     1539
##      RRD SR Slow Rst     Pork Belly 150g
##                                     1526
##   Smiths Thinly Cut   Roast Chicken 175g
##                                     1519
##      RRD Sweet Chilli &  Sour Cream 165g
##                                     1516
##           Woolworths Cheese   Rings 190g
##                                     1516
##                         CCs Original 175g
##                                     1514
##         RRD Honey Soy       Chicken 165g
##                                     1513
##    Smith Crinkle Cut   Mac N Cheese 150g
```

```
##                                     1512
##      WW Supreme Cheese   Corn Chips 200g
##                                     1509
##  Infuzions Mango     Chutny Papadums 70g
##                                     1507
##         RRD Chilli&         Coconut 150g
##                                     1506
##      Smiths Crinkle Cut  Snag&Sauce 150g
##                                     1503
##                 CCs Nacho Cheese    175g
##                                     1498
##  Red Rock Deli Sp    Salt & Truffle 150G
##                                     1498
##     Red Rock Deli Thai  Chilli&Lime 150g
##                                     1495
##           WW Original Corn    Chips 200g
##                                     1495
##           Woolworths Mild     Salsa 300g
##                                     1491
##  Smiths Crinkle Cut  Chips Barbecue 170g
##                                     1489
##           WW Original Stacked Chips 160g
##                                     1487
##   Smiths Crinkle Cut  Chips Chicken 170g
##                                     1484
##   WW Sour Cream &OnionStacked Chips 160g
##                                     1483
##  Smiths Crinkle Cut  Chips Chs&Onion170g
##                                     1481
##           Cheetos Chs & Bacon Balls 190g
##                                     1479
##                 RRD Salt & Vinegar  165g
##                                     1474
##                 RRD Lime & Pepper   165g
##                                     1473
##   Smiths Chip Thinly  S/Cream&Onion 175g
##                                     1473
##                 Doritos Salsa Mild  300g
##                                     1472
##    Smiths Crinkle Cut  Tomato Salsa 150g
##                                     1470
##        WW D/Style Chip     Sea Salt 200g
##                                     1469
##    GrnWves Plus Btroot & Chilli Jam 180g
##                                     1468
##   Natural Chip        Compny SeaSalt175g
##                                     1468
##         WW Crinkle Cut      Chicken 175g
##                                     1467
##  Smiths Crinkle Cut  Chips Original 170g
##                                     1461
## Smiths Thinly       Swt Chli&S/Cream175G
##                                     1461
##   Natural ChipCo      Hony Soy Chckn175g
```

```
##                                     1460
## Red Rock Deli SR    Salsa & Mzzrlla 150g
##                                     1458
##     RRD Steak &         Chimuchurri 150g
##                                     1455
##                                   (Other)
##                                    21550
```

```
summary(trans)
```

```
##       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.:  70021   1st Qu.:  67602
##  Median :2018-12-30   Median :130.0   Median : 130358   Median : 135138
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135550   Mean   : 135158
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202701
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
##
##     PROD_NBR       PROD_NAME            PROD_QTY         TOT_SALES
##  Min.   :  1.00   Length:264836      Min.   :  1.000   Min.   :  1.500
##  1st Qu.: 28.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.400
##  Median : 56.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.58                      Mean   :  1.907   Mean   :  7.304
##  3rd Qu.: 85.00                      3rd Qu.:  2.000   3rd Qu.:  9.200
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
##
##                                  PROD_NAME_FACTOR
##  Kettle Mozzarella   Basil & Pesto 175g  :  3304
##  Kettle Tortilla ChpsHny&Jlpno Chili 150g:  3296
##  Cobs Popd Swt/Chlli &Sr/Cream Chips 110g:  3269
##  Tyrrells Crisps     Ched & Chives 165g  :  3268
##  Cobs Popd Sea Salt  Chips 110g          :  3265
##  Kettle 135g Swt Pot Sea Salt            :  3257
##  (Other)                                 :245177
```

```
str(trans)
```

```
## tibble [264,836 x 9] (S3: tbl_df/tbl/data.frame)
##  $ DATE            : Date[1:264836], format: "2018-10-17" "2019-05-14" ...
##  $ STORE_NBR       : num [1:264836] 1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR  : num [1:264836] 1000 1307 1343 2373 2426 ...
##  $ TXN_ID          : num [1:264836] 1 348 383 974 1038 ...
##  $ PROD_NBR        : num [1:264836] 5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME       : chr [1:264836] "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175
##  $ PROD_QTY        : num [1:264836] 2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES       : num [1:264836] 6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...
##  $ PROD_NAME_FACTOR: Factor w/ 114 levels "Burger Rings 220g",..: 44 2 80 76 43 51 78 23 14 24 ...
```

Text analysis Looks like we are definitely looking at potato chips but how can we check that these are all chips? We can do some basic text analysis by summarising the individual words in the product name.

```
productWords <- data.table(unlist(strsplit(unique(trans$PROD_NAME), " ")))
setnames(productWords, 'words')
```

Examine product words in PROD_NAME As we are only interested in words that will tell us if the product is chips or not, let's remove all words with digits and special characters such as '&' from our set of product words. We can do this using grepl().

```
numerical.validation <- grepl('[1-9]',productWords[,words])
productWords <- productWords[numerical.validation==FALSE]
#productWords
```

Removing words that contain numerical

```
scAnd.validation <- grepl('&',productWords[,words])
productWords <- productWords[scAnd.validation==FALSE]
```

Removing words that contain special character '&'

```
sc.validation <- grepl('/',productWords[,words])
productWords <- productWords[sc.validation==FALSE]
```

Removing words that contain special character '/'

```
#factorising words
productWords <- factor(productWords$words)
```

Counting frequencies summary

```
summary(productWords)
```

```
##                 Chips      Smiths     Crinkle         Cut      Kettle
##         234        21          16          14          14          13
##      Cheese      Salt    Original        Chip     Doritos       Salsa
##          12        12          10           9           9           9
##        Corn    Pringles        RRD     Chicken          WW         Sea
##           8         8           8           7           7           6
##        Sour     Chilli      Crisps      Thinly       Thins     Vinegar
##           6         5           5           5           5           5
##       Cream      Deli    Infuzions     Natural         Red        Rock
##           4         4           4           4           4           4
##     Supreme       CCs        Cobs         Dip          El        Lime
##           4         3           3           3           3           3
##        Mild       Old        Paso        Popd  Sensations         Soy
##           3         3           3           3           3           3
##       Sweet     Tomato    Tortilla    Tostitos    Twisties  Woolworths
##           3         3           3           3           3           3
##         And       BBQ      Burger     Cheetos    Cheezels      ChipCo
##           2         2           2           2           2           2
##      Chives     French       Grain       Honey     Lightly      Medium
##           2         2           2           2           2           2
##       Nacho     Potato       Rings      Salted       Smith          SR
##           2         2           2           2           2           2
##         Swt      Tangy        Thai    Tyrrells       Waves       Aioli
##           2         2           2           2           2           1
```

```
##       Bacon         Bag       Balls    Barbecue    Barbeque       Basil
##           1           1           1           1           1           1
##       Belly         Big   Bolognese         Box      Btroot   Camembert
##           1           1           1           1           1           1
##        Ched       Chili Chimuchurri    Chipotle       Chnky         Chp
##           1           1           1           1           1           1
##         Chs      Chutny          Co     Coconut      Compny    Crackers
##           1           1           1           1           1           1
##       Crips         Crm         Crn     (Other)
##           1           1           1          70
```

```
trans <- data.table(trans)
```

There are salsa products in the dataset but we are only interested in the chips category, so let's remove these.
Remove salsa products

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

```
summary(trans)
```

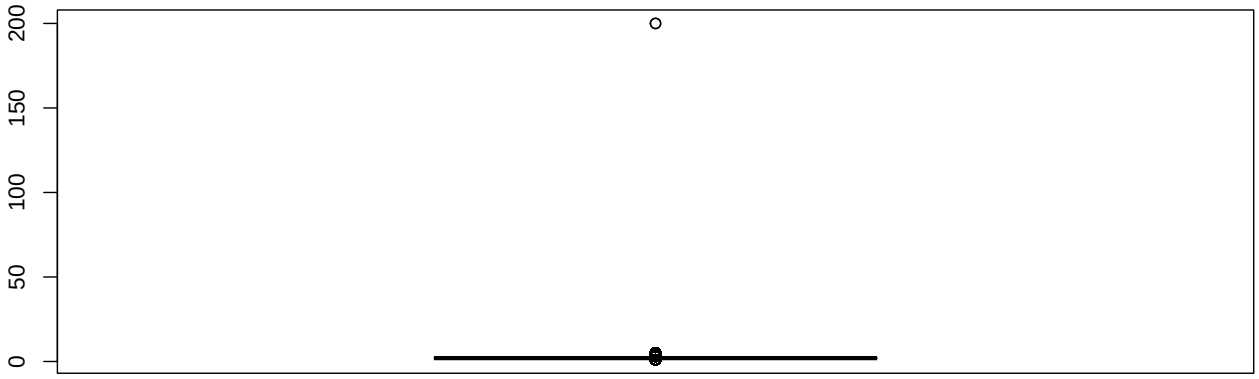
Summarise the data to check for nulls and possible outliers

```
##       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
##
##                                  PROD_NAME_FACTOR
##  Kettle Mozzarella   Basil & Pesto 175g  :  3304
##  Kettle Tortilla ChpsHny&Jlpno Chili 150g:  3296
##  Cobs Popd Swt/Chlli &Sr/Cream Chips 110g:  3269
##  Tyrrells Crisps     Ched & Chives 165g  :  3268
##  Cobs Popd Sea Salt  Chips 110g          :  3265
##  Kettle 135g Swt Pot Sea Salt            :  3257
##  (Other)                                 :227083
```

Checking for outliers By seeing summary of data, we can see that the maximum value of PROD_QTY is more that (3rd quartile + 1.5*IQR)

Lets confirm this with a boxplot

```
boxplot(trans$PROD_QTY)
```



Yes we can confirm existence of outliers.

Let's investigate further the case where 200 packets of chips are bought in one transaction.

```
filter(trans,trans$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
##                     PROD_NAME_FACTOR
## 1: Dorito Corn Chp     Supreme 380g
## 2: Dorito Corn Chp     Supreme 380g
```

We have 2 records where the PROD_QTY is 200. Both are made by same customer 226000.

Let's see if he has any other transactions

```
filter(trans,trans$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
##                     PROD_NAME_FACTOR
## 1: Dorito Corn Chp     Supreme 380g
## 2: Dorito Corn Chp     Supreme 380g
```

It looks like this customer has only had the two transactions over the year and is not an ordinary retail customer. The customer might be buying chips for commercial purposes instead. We'll remove this loyalty card number from further analysis.

Filter out the customer based on the loyalty card number Removing customer - 226000 from further analysis

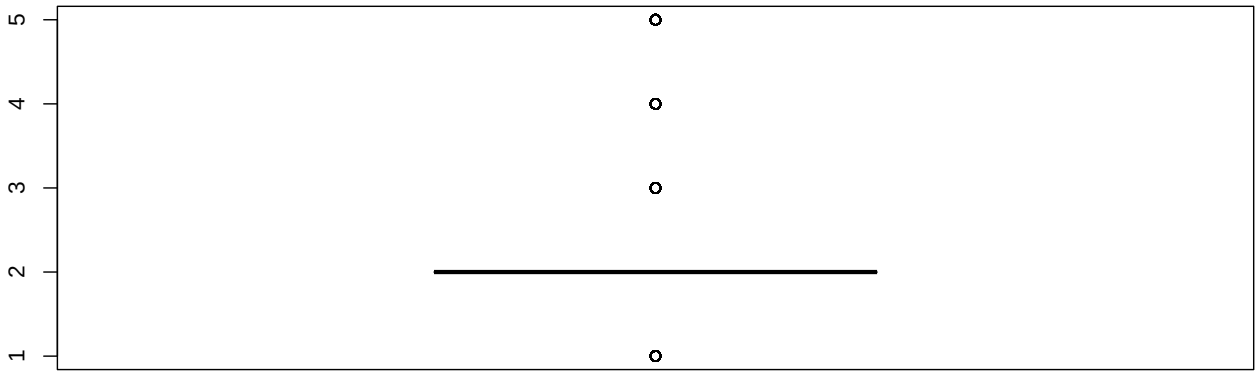
```
#trans[,trans$LYLTY_CARD_NBR != 226000]
trans <- trans[trans[,trans$LYLTY_CARD_NBR != 226000]]
```

```
summary(trans)
```

Re-examine transaction data

```
##       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
##
##                                  PROD_NAME_FACTOR
##  Kettle Mozzarella   Basil & Pesto 175g  :  3304
##  Kettle Tortilla ChpsHny&Jlpno Chili 150g:  3296
##  Cobs Popd Swt/Chlli &Sr/Cream Chips 110g:  3269
##  Tyrrells Crisps     Ched & Chives 165g  :  3268
##  Cobs Popd Sea Salt  Chips 110g          :  3265
##  Kettle 135g Swt Pot Sea Salt            :  3257
##  (Other)                                 :227081
```

```
boxplot(trans$PROD_QTY)
```



Count the number of transactions by date Let us factorise the dates

```
trans$newDATE <- factor(trans$DATE)
```

Summary

```
summary(trans)
```

```
##       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
##
##                                  PROD_NAME_FACTOR        newDATE
##  Kettle Mozzarella   Basil & Pesto 175g  :  3304   2018-12-24:   865
##  Kettle Tortilla ChpsHny&Jlpno Chili 150g:  3296   2018-12-23:   853
##  Cobs Popd Swt/Chlli &Sr/Cream Chips 110g:  3269   2018-12-22:   840
##  Tyrrells Crisps     Ched & Chives 165g  :  3268   2018-12-19:   839
##  Cobs Popd Sea Salt  Chips 110g          :  3265   2018-12-20:   808
##  Kettle 135g Swt Pot Sea Salt            :  3257   2018-12-18:   799
##  (Other)                                 :227081   (Other)   :241736
```

```
str(trans)
```

```
## Classes 'data.table' and 'data.frame':   246740 obs. of  10 variables:
##  $ DATE            : Date, format: "2018-10-17" "2019-05-14" ...
##  $ STORE_NBR       : num  1 1 1 2 2 4 4 5 7 7 ...
##  $ LYLTY_CARD_NBR  : num  1000 1307 1343 2373 2426 ...
##  $ TXN_ID          : num  1 348 383 974 1038 ...
##  $ PROD_NBR        : num  5 66 61 69 108 16 24 42 52 16 ...
##  $ PROD_NAME       : chr  "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g" "Smiths
##  $ PROD_QTY        : num  2 3 2 5 3 1 1 1 2 1 ...
##  $ TOT_SALES       : num  6 6.3 2.9 15 13.8 5.7 3.6 3.9 7.2 5.7 ...
##  $ PROD_NAME_FACTOR: Factor w/ 114 levels "Burger Rings 220g",..: 44 2 80 76 43 78 23 14 24 78 ...
##  $ newDATE         : Factor w/ 364 levels "2018-07-01","2018-07-02",..: 109 317 323 48 49 319 319 51
##  - attr(*, ".internal.selfref")=<externalptr>
```

There are 364 unique dates where transaction happened. We will create a new column with dates from min to max i.e., 2018-07-01 to 2019-06-30 and then join this with trans to find that missing date.

```
model.date <- seq(as.Date("2018-07-01"),as.Date("2019-06-30"),by = 'day')
model.date <- data.table(model.date)
setnames(model.date,'DATE')
#colnames(model.date) <- c('Date')
```

Full-join Date

```
trans <- full_join(trans,model.date,by = c('DATE'))
```

summary

```
summary(trans)
```

```
##       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
##                       NA's   :1       NA's   :1         NA's   :1
```

```
##     PROD_NBR       PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:246741      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
##  NA's   :1                           NA's   :1       NA's   :1
##                                  PROD_NAME_FACTOR        newDATE
##  Kettle Mozzarella   Basil & Pesto 175g  :  3304   2018-12-24:   865
##  Kettle Tortilla ChpsHny&Jlpno Chili 150g:  3296   2018-12-23:   853
##  Cobs Popd Swt/Chlli &Sr/Cream Chips 110g:  3269   2018-12-22:   840
##  Tyrrells Crisps     Ched & Chives 165g  :  3268   2018-12-19:   839
##  Cobs Popd Sea Salt  Chips 110g          :  3265   2018-12-20:   808
##  (Other)                                 :230338   (Other)   :242535
##  NA's                                    :     1   NA's      :     1
```

finding the date

```
filter(trans,is.na(trans$STORE_NBR) == TRUE)
```

```
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME PROD_QTY
## 1: 2018-12-25        NA             NA     NA       NA      <NA>       NA
##    TOT_SALES PROD_NAME_FACTOR newDATE
## 1:        NA             <NA>    <NA>
```

We can see that the date 2018-12-25 is missing.

```
transactions_per_date <- trans[, as.Date(trans$DATE, format = "%Y-%m-%d")]
transactions_per_date <- table(transactions_per_date)
transactions_per_date <- data.table(transactions_per_date)
```

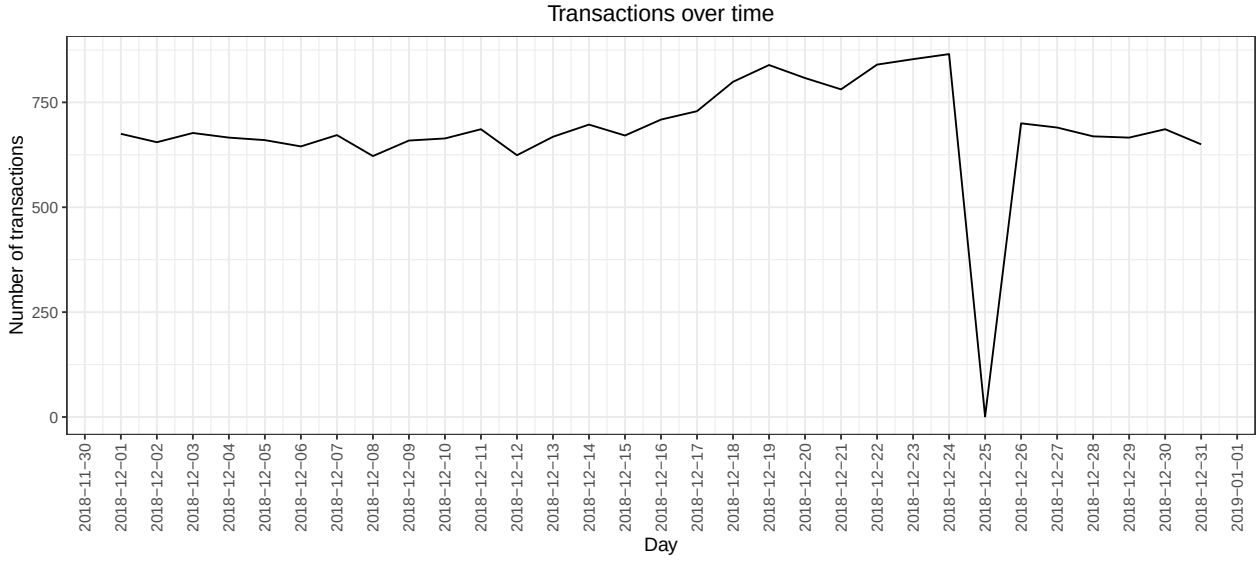
Count the number of transactions by date

```
#### Setting plot themes to format graphs
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
#### Plot transactions over time
ggplot(transactions_per_date, aes(x = as.Date(transactions_per_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))
```

Plot transactions over time

Filter to December and look at individual days We can see some anomaly in December. Creating December chart to further investigate.

```
x = subset(transactions_per_date, format.Date(transactions_per_date,"%m")=="12")
ggplot(x, aes(x = as.Date(transactions_per_date), y = N)) +
 geom_line() +
 labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +
 scale_x_date(breaks = "1 day") +
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



We can see that on Dec 25 we do not have any transaction. Because it was a missing value. Sales got increased until Christmas day and on Christmas day shops were closed.

```
# removing the Christmas day
trans <- subset(trans,trans$DATE != '2018-12-25')
```

Feature Engineering

Now that we are satisfied that the data no longer has outliers, we can move on to creating other features such as brand of chips or pack size from PROD_NAME. We will start with pack size.

```
# creating new Pack size feature in trans by parasing our numbers from product names.
trans[, PACK_SIZE := parse_number(PROD_NAME)]
# Always check your output
# Let's check if the pack sizes look sensible
#.N is a spl variable in data.table used to represent # of observations in a group along with by = pack
trans[, .N, PACK_SIZE][order(PACK_SIZE)]
```

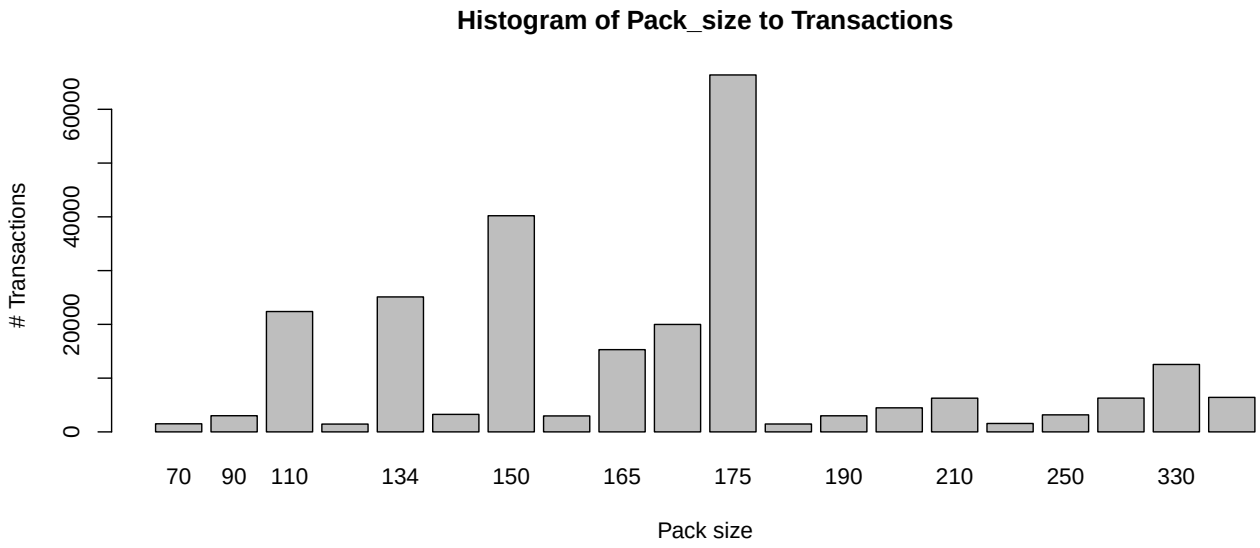
Creating new features: Pack size

```
##     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
```

The largest size is 380g and the smallest size is 70g - seems sensible!

```
x1 <- trans$PACK_SIZE
x1 <- table(x1)
x1<- data.table(x1)
colnames(x1) <- c('Pack_size','Transactions')

barplot(height = x1$Transactions,
        names.arg = x1$Pack_size,
        main="Histogram of Pack_size to Transactions",
        xlab = "Pack size",
        ylab= "# Transactions")
```

Histogram of Pack size.



Creating new features: Brand_name Pack sizes created look reasonable and now to create brands, we can use the first word in PROD_NAME to work out the brand name

```
#Here we are parsing the first word of the sentence using word() from stringr
trans$Brand_name <- word(trans$PROD_NAME, 1)

#checking brands results
trans[, .N, Brand_name][order(Brand_name)]
```

```
##     Brand_name     N
##  1:     Burger  1564
##  2:        CCs  4551
##  3:    Cheetos  2927
##  4:   Cheezels  4603
##  5:       Cobs  9693
##  6:     Dorito  3183
##  7:    Doritos 22041
##  8:     French  1418
##  9:      Grain  6272
## 10:    GrnWves  1468
## 11:  Infuzions 11057
## 12:     Infzns  3144
## 13:     Kettle 41288
## 14:        NCC  1419
## 15:    Natural  6050
## 16:   Pringles 25102
## 17:        RRD 11894
## 18:        Red  4427
## 19:      Smith  2963
## 20:     Smiths 27390
## 21:      Snbts  1576
## 22:   Sunbites  1432
## 23:      Thins 14075
## 24:   Tostitos  9471
## 25:   Twisties  9454
## 26:   Tyrrells  6442
## 27:         WW 10320
```

```
## 28: Woolworths  1516
##     Brand_name     N
```

Clean brand names Some of the brand names look like they are of the same brands - such as RED and RRD, NCC and Natural Chip Co, Smith and Smiths, infuzions and infzns, Snbts and Sunbites, WW and Woolworths, Dorito and Doritos, Grain and GrnWves Let's combine these together.

```
#clean brand names
trans[Brand_name == "Red", Brand_name := "RRD"]
trans[Brand_name == "Dorito", Brand_name := "Doritos"]
trans[Brand_name == "Grain", Brand_name := "GrnWves"]
trans[Brand_name == "NCC", Brand_name := "Natural"]
trans[Brand_name == "Smith", Brand_name := "Smiths"]
trans[Brand_name == "Infzns", Brand_name := "Infuzions"]
trans[Brand_name == "Snbts", Brand_name := "Sunbites"]
trans[Brand_name == "WW", Brand_name := "Woolworths"]
```

```
#checking brands results
trans[, .N, Brand_name][order(Brand_name)]
```

```
##     Brand_name     N
##  1:     Burger  1564
##  2:        CCs  4551
##  3:    Cheetos  2927
##  4:   Cheezels  4603
##  5:       Cobs  9693
##  6:    Doritos 25224
##  7:     French  1418
##  8:    GrnWves  7740
##  9:  Infuzions 14201
## 10:     Kettle 41288
## 11:    Natural  7469
## 12:   Pringles 25102
## 13:        RRD 16321
## 14:     Smiths 30353
## 15:   Sunbites  3008
## 16:      Thins 14075
## 17:   Tostitos  9471
## 18:   Twisties  9454
## 19:   Tyrrells  6442
## 20: Woolworths 11836
```

```
cust = read.csv('QVI_purchase_behaviour.csv')
```

Loading customer dataset

Examining customer data

```
summary(cust)
```

```
##  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
```

```
str(cust)
```

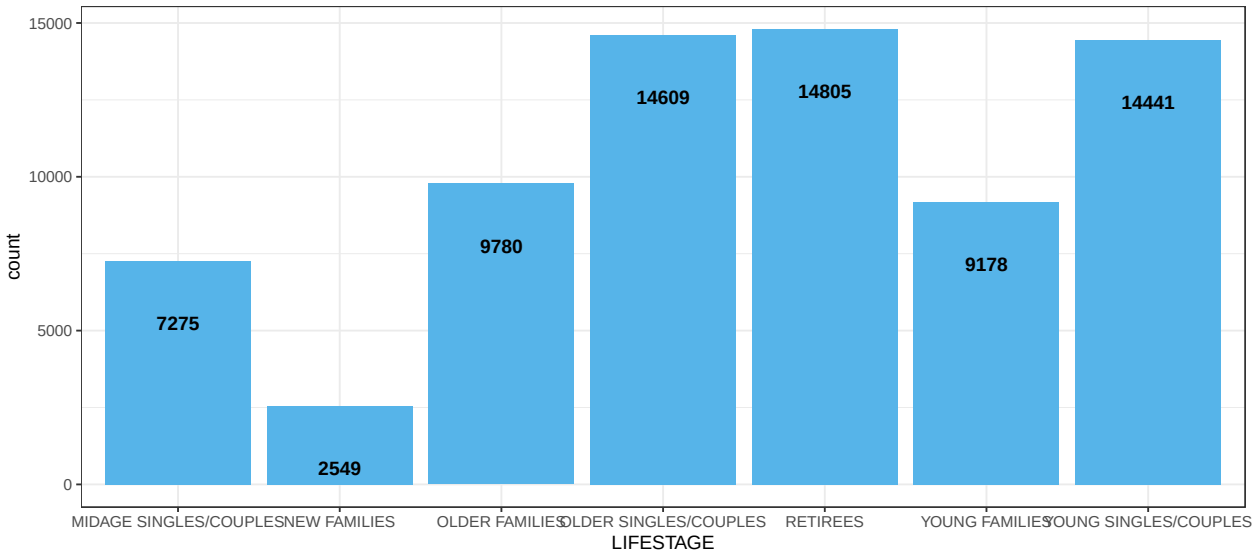
```
## 'data.frame':    72637 obs. of  3 variables:
##  $ LYLTY_CARD_NBR  : int  1000 1002 1003 1004 1005 1007 1009 1010 1011 1012 ...
##  $ LIFESTAGE       : chr  "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "OLDER SI
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream"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

Let's have a closer look at the LIFESTAGE and PREMIUM_CUSTOMER columns.

```
#distribution of lifestage and premium_customer
ggplot(data = cust, aes(x = LIFESTAGE)) +
  geom_histogram(stat = "count", fill = "#56B4E9") +
  geom_text(stat = "count", aes(label = ..count..), fontface = "bold", vjust = 5)
```

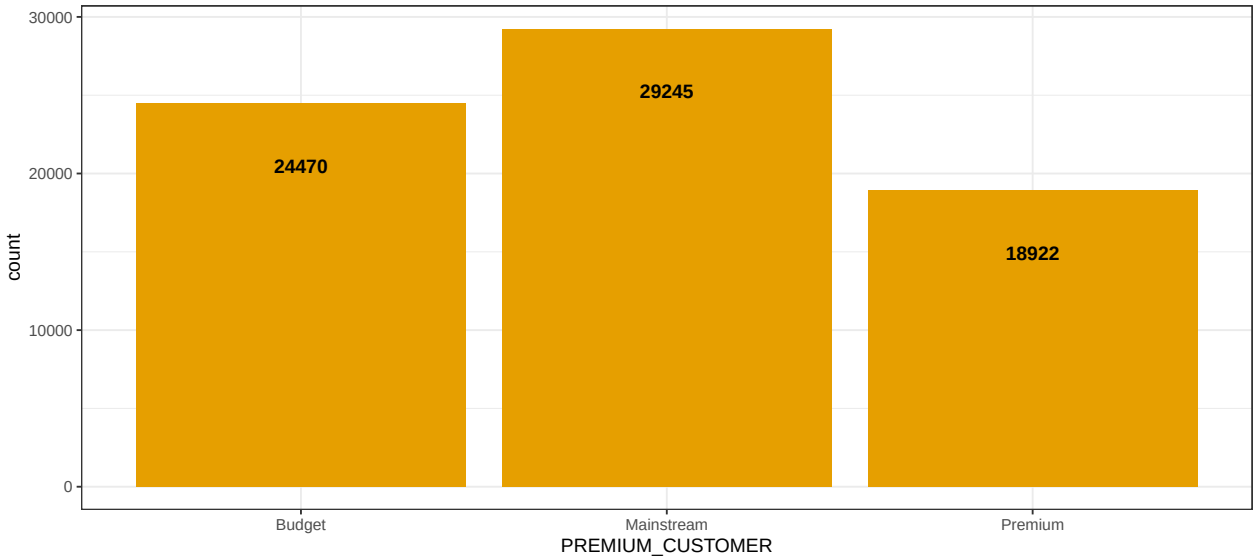
```
## Warning in geom_histogram(stat = "count", fill = "#56B4E9"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```

```
## Warning: The dot-dot notation (`..count..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(count)` instead.
```




```
ggplot(data = cust,aes(x = PREMIUM_CUSTOMER)) +
  geom_histogram(stat = "count", fill = "#E69F00") +
  geom_text(stat = "count",aes(label = ..count..), fontface = "bold",vjust = 5)
```

```
## Warning in geom_histogram(stat = "count", fill = "#E69F00"): Ignoring unknown
## parameters: `binwidth`, `bins`, and `pad`
```




```
#### Examining the values of lifestage and premium_customer(same as above but just numerical values)
cust %>%
  group_by(LIFESTAGE) %>%
  summarise(cust_count = n()) %>%
  arrange(desc(cust_count))
```

```
## # A tibble: 7 x 2
##   LIFESTAGE              cust_count
##   <chr>                       <int>
## 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
```

```
cust %>%
  group_by(PREMIUM_CUSTOMER) %>%
  summarise(prem_count = n()) %>%
  arrange(desc(prem_count))
```

```
## # A tibble: 3 x 2
##   PREMIUM_CUSTOMER prem_count
##   <chr>                 <int>
## 1 Mainstream            29245
## 2 Budget                24470
## 3 Premium               18922
```

As there do not seem to be any issues with the customer data, we can now go ahead and join the transaction and customer data sets together

```
#### Merge transaction data to customer data
# all.x = T implies full left join
df <- merge(trans, cust, all.x = TRUE)
```

```
#df$LYLTY_CARD_NBR <- as.factor(df$LYLTY_CARD_NBR)
```

Merge transaction data to customer data As the number of rows in `data` is the same as that of `transactionData`, we can be sure that no duplicates were created. This is because we created `data` by setting `all.x = TRUE` (in other words, a left join) which means take all the rows in `transactionData` and find rows with matching values in shared columns and then joining the details in these rows to the `x` or the first mentioned table.

Checking if any transactions did not have a matched customer.

```
df[is.null(LIFESTAGE)]
```

```
## Empty data.table (0 rows and 14 cols): LYLTY_CARD_NBR,DATE,STORE_NBR,TXN_ID,PROD_NBR,PROD_NAME...
```

```
df[is.null(PREMIUM_CUSTOMER)]
```

```
## Empty data.table (0 rows and 14 cols): LYLTY_CARD_NBR,DATE,STORE_NBR,TXN_ID,PROD_NBR,PROD_NAME...
```

Great, there are no nulls! So all our customers in the transaction data has been accounted for in the customer dataset.

```
#Not adding DEC 25
df <- df[!DATE == '2018-12-25']
```

```
filePath <- "/Users/santosh/Documents/QuantiumDA/quantiumDA/"
fwrite(df, paste0(filePath,"QVI_data.csv"))
```

Saving dataset Data exploration is now complete!

Exploratory data analysis on customer segments

Now that the data is ready for analysis, we can define some metrics of interest to the client: • Who spends the most on chips (total sales), describing customers by lifestage and how premium their general purchasing behavior is • How many customers are in each segment • How many chips are bought per customer by segment • What's the average chip price by customer segment We could also ask our data team for more information. Examples are: • The customer's total spend over the period and total spend for each transaction to understand what proportion of their grocery spend is on chips • Proportion of customers in each customer segment overall to compare against the mix of customers who purchase chips.

Let's start with calculating total sales by LIFESTAGE and PREMIUM_CUSTOMER and plotting the split by these segments to describe which customer segment contribute most to chip sales.

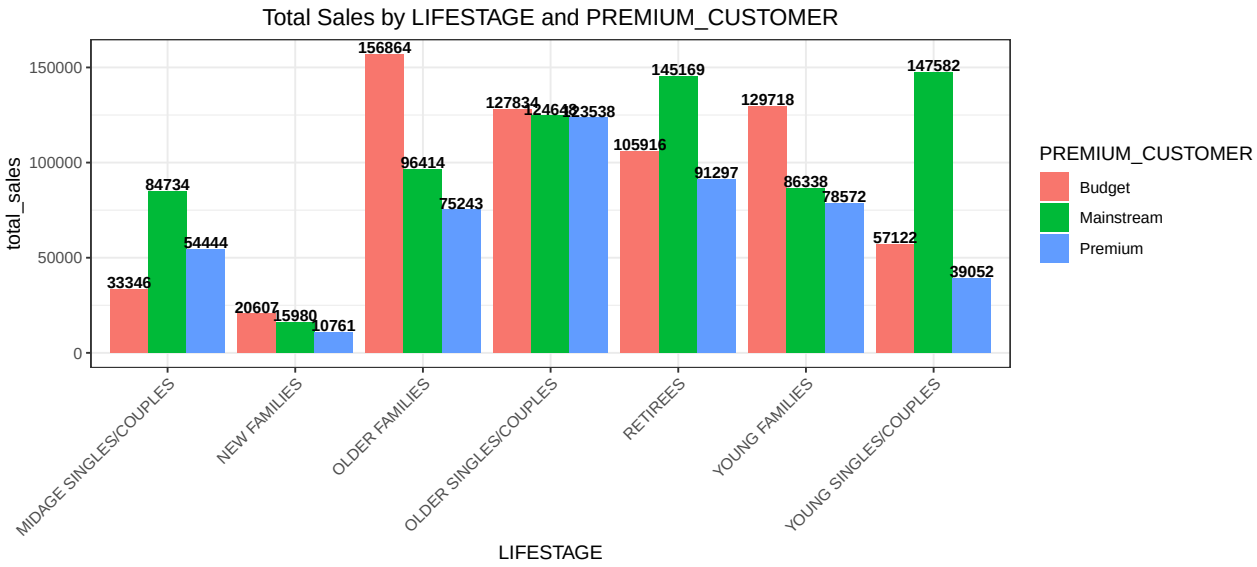
```
#salesxxx <- df[, .(SALES = sum(TOT_SALES)), .(LIFESTAGE, PREMIUM_CUSTOMER)]
#same method different approach, upper uses data.table functions, below uses dplyr package

sales_summary <- df %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(total_sales = sum(TOT_SALES))
```

Total sales by LIFESTAGE and PREMIUM_CUSTOMER

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

```
ggplot(sales_summary, aes(x = LIFESTAGE, y = total_sales, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = round(total_sales)), position = position_dodge(width = 0.9),fontface = "bold", v
```

```
  labs(title = "Total Sales by LIFESTAGE and PREMIUM_CUSTOMER") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



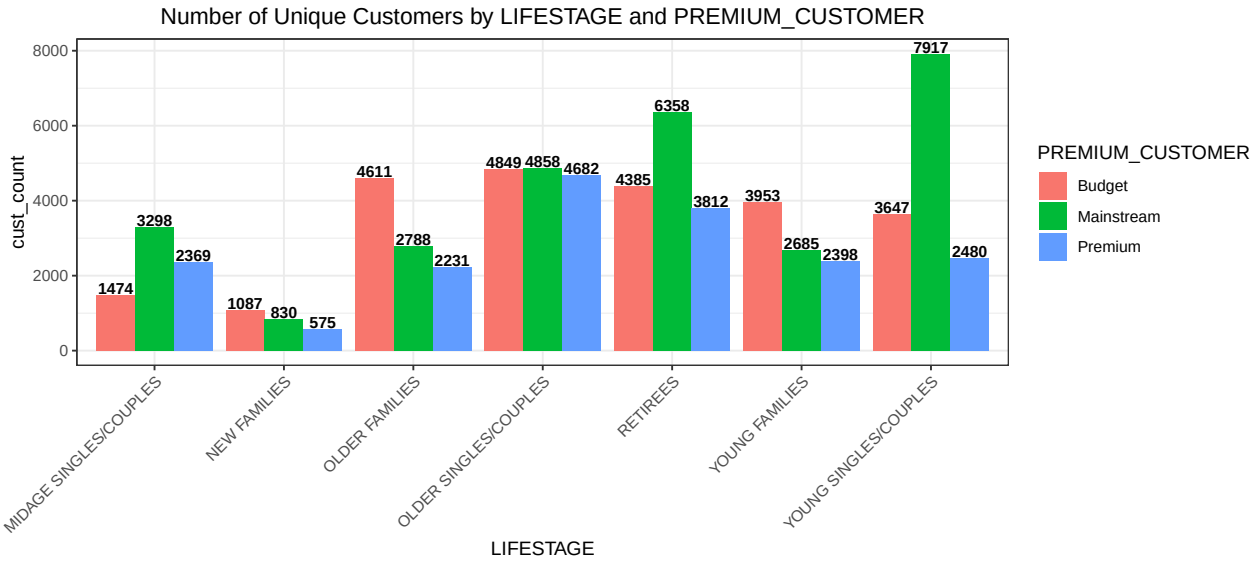
Sales are coming mainly from Budget - older families, Mainstream - young singles/couples, and Mainstream - retirees

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

```
#count distinct values of customers
cust_summary <- df %>%
  distinct(LYLTY_CARD_NBR, .keep_all = TRUE) %>%
  count(LIFESTAGE, PREMIUM_CUSTOMER, name = "cust_count")
```

```
ggplot(cust_summary, aes(x = LIFESTAGE, y = cust_count, fill = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge")  +
  geom_text(aes(label = round(cust_count)), position = position_dodge(width = 0.9),fontface = "bold", v
  labs(title = "Number of Unique Customers by LIFESTAGE and PREMIUM_CUSTOMER") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Number of customers by LIFESTAGE and PREMIUM_CUSTOMER



There are more Mainstream - young singles/couples and Mainstream - retirees who buy chips. This contributes to there being more sales to these customer segments but this is not a major driver for the Budget - Older families segment.

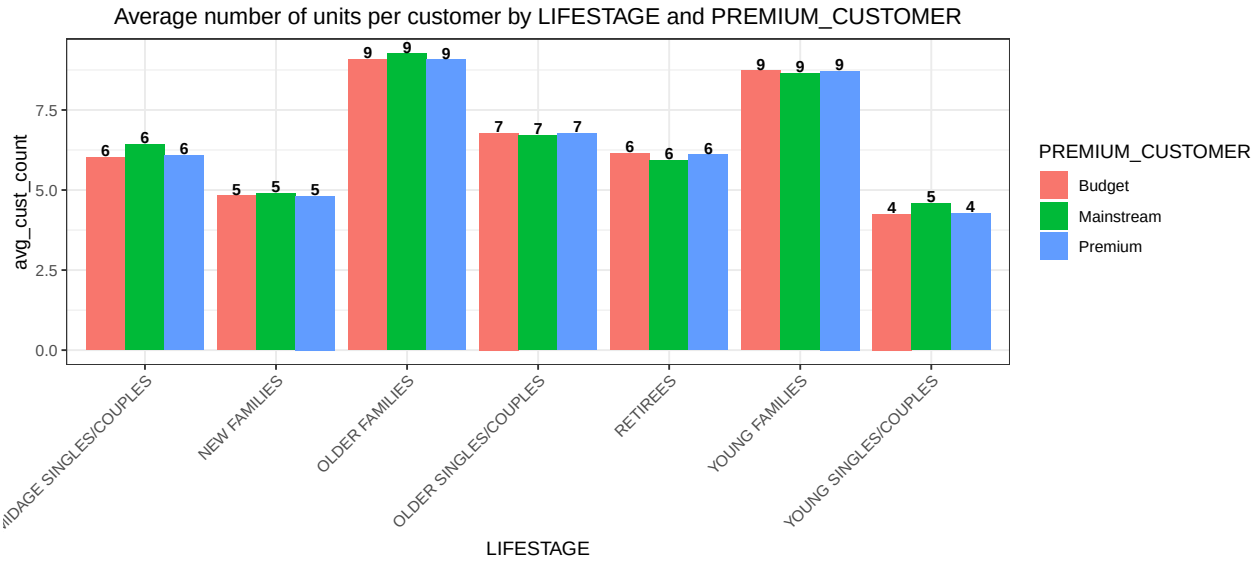
Higher sales may also be driven by more units of chips being bought per customer. Let's have a look at this next.

```
avg_cust <- df %>% group_by(LIFESTAGE , PREMIUM_CUSTOMER ) %>%
  summarize(avg_cust_count = sum(PROD_QTY)/n_distinct(LYLTY_CARD_NBR))
```

Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER

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

```
#### Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER
ggplot(data = avg_cust,aes(x = LIFESTAGE,y = avg_cust_count,fill = PREMIUM_CUSTOMER)) +   geom_bar(stat
  geom_text(aes(label = round(avg_cust_count)),fontface = "bold",vjust = -0.1, size = 3, position = posi
  labs(title = "Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Older families and young families in general buy more chips per customer.

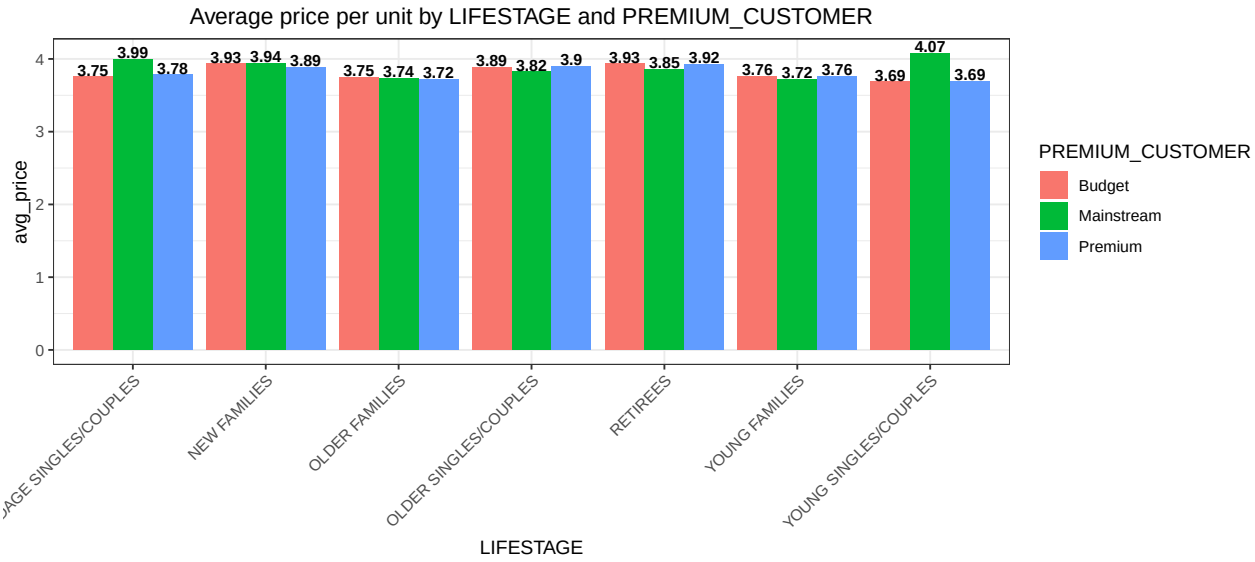
Let's also investigate the average price per unit chips bought for each customer segment as this is also a driver of total sales.

```
avg_ppu <- df %>% group_by(LIFESTAGE , PREMIUM_CUSTOMER) %>%
  summarise(avg_price = (sum(TOT_SALES)/sum(PROD_QTY)))
```

Average price per unit by LIFESTAGE and PREMIUM_CUSTOMER

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

```
#### Average price per unit by LIFESTAGE and PREMIUM_CUSTOMER
ggplot(data = avg_ppu,aes(x = LIFESTAGE,y = avg_price,fill = PREMIUM_CUSTOMER)) +   geom_bar(stat = "id
  geom_text(aes(label = round(avg_price,2)), position = position_dodge(width = 0.9),fontface = "bold",v
  labs(title = "Average price per unit by LIFESTAGE and PREMIUM_CUSTOMER") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Mainstream midage and young singles and couples are more willing to pay more per packet of chips compared to their budget and premium counterparts. This may be due to premium shoppers being more likely to buy healthy snacks and when they buy chips, this is mainly for entertainment purposes rather than their own consumption. This is also supported by there being fewer premium midage and young singles and couples buying chips compared to their mainstream counterparts.

Perform an independent t-test Here we are performing t-test between Mainstream vs (Premium or Budget) for Midage and young - singles and couples.

```
#### Perform an independent t-test between mainstream vs premium and budget midage and young singles an
# Over to you! Perform a t-test to see if the difference is significant.
main_premium <- subset(avg_ppu, PREMIUM_CUSTOMER %in% c("Mainstream", "Premium")&LIFESTAGE %in% c("YOUN

t.test(avg_price~PREMIUM_CUSTOMER, data = main_premium)
```

```
##
##  Welch Two Sample t-test
##
## data:  avg_price by PREMIUM_CUSTOMER
## t = 5.0147, df = 1.9805, p-value = 0.0383
## alternative hypothesis: true difference in means between group Mainstream and group Premium is not e
## 95 percent confidence interval:
##  0.03980094 0.55497920
## sample estimates:
## mean in group Mainstream    mean in group Premium
##                 4.034246                 3.736856
```

```
#### Perform an independent t-test between mainstream vs premium and budget midage and young singles an
# Over to you! Perform a t-test to see if the difference is significant.
main_budget <- subset(avg_ppu, PREMIUM_CUSTOMER %in% c("Mainstream","Budget")&LIFESTAGE %in% c("YOUNG SI

t.test(avg_price~PREMIUM_CUSTOMER, data = main_budget)
```

```
##
##  Welch Two Sample t-test
##
## data:  avg_price by PREMIUM_CUSTOMER
## t = -5.9898, df = 1.9572, p-value = 0.02815
## alternative hypothesis: true difference in means between group Budget and group Mainstream is not equ
## 95 percent confidence interval:
##  -0.5454862 -0.0838308
## sample estimates:
##     mean in group Budget mean in group Mainstream
##                 3.719587                 4.034246
```

```
t.test(avg_ppu[avg_ppu$LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & avg_ppu$PRI
       avg_ppu[avg_ppu$LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & avg_ppu$PRI
       alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  avg_ppu[avg_ppu$LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & avg_ppu$
## t = 6.6358, df = 1.7353, p-value = 0.01556
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
```

```
##  0.1562509       Inf
## sample estimates:
## mean of x mean of y
##  4.034246  3.728222
```

All the t-test results suggest that p-value is less than $\alpha = 0.05$ and we accept alternate hypothesis that there is some difference between mean between the groups mainstream vs premium or budget.

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 Mainstream, young singles/couples We might want to target customer segments that contribute the most to sales to retain them or further increase sales. Let's look at Mainstream - young singles/couples. For instance, let's find out if they tend to buy a particular brand of chips.

```
#### Deep dive into Mainstream, young singles/couples
segment1 <- df[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER == "Mainstream",]
other <- df[!(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_name]
brand_proportions <- merge(quantity_segment1_by_brand, quantity_other_by_brand)[, affinityToBrand := ta
brand_proportions[order(-affinityToBrand)]
```

```
##     Brand_name 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 22% 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_SI
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 := targetS
pack_proportions[order(-affinityToPack)]
```

Preferred pack size compared to the rest of the population

```
##     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.

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

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

```
df[Brand_name == "Tyrrells", unique(PROD_NAME)]
```

```
## [1] "Tyrrells Crisps     Ched & Chives 165g"
## [2] "Tyrrells Crisps     Lightly Salted 165g"
```

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

Even though Tyrrells is top likely brand for Mainstream - young singles/couples, they are 11% less likely to purchase 165g packets.

Conclusion 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.

With these Insights, my task-1 is completed.

End of Task-1