Quantium Virtual Internship - Retail Strategy and Analytics - Task 1 Krutarth Patel

Solution Task 1

This file is a solution for the Task 1 of the Quantium Virtual Internship. It will walk you through the analysis.

Load required libraries and datasets

Note that you will need to install these libraries if you have never used these before.

```
library(data.table)
library(ggplot2)
library(ggmosaic)
library(readr)
library(readxl)
library(tidyverse)
library(dplyr)
transactionData <- read_excel("~/Quantium Internship/QVI_transaction_data.xlsx")
customerData <- QVI_purchase_behaviour <- read_csv("~/Quantium
↪  Internship/QVI_purchase_behaviour.csv")
```

Exploratory data analysis

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

Examining transaction data

We can use `str()` to look at the format of each column and see a sample of the data. As we have read in the dataset as a `data.table` object, we can also run `transactionData` in the console to see a sample of the data or use `head(transactionData)` to look at the first 10 rows. Let's check if columns we would expect to be numeric are in numeric form and date columns are in date format.

```
#### Examine transaction data
str(transactionData)
```

```
## 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" "Smiths Crinkle Cut Chips Chicken 170g" "Smiths Chip Thinly
S/Cream&Onion 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 ...
```

We can see that the date column is in an integer format. Let's change this to a date format. A quick search online tells us that CSV and Excel integer dates begin on 30 Dec 1899

```
#### Converting DATE column to a date format
transactionData$DATE <- as.Date(transactionData$DATE, origin = "1899-12-30")
```

We should check that we are looking at the right products by examining PROD_NAME.

```
unique(transactionData$PROD_NAME)
```

```
##   [1] "Natural Chip        Compny SeaSalt175g"
##   [2] "CCs Nacho Cheese    175g"
##   [3] "Smiths Crinkle Cut  Chips Chicken 170g"
##   [4] "Smiths Chip Thinly  S/Cream&Onion 175g"
##   [5] "Kettle Tortilla ChpsHny&Jlpno Chili 150g"
##   [6] "Old El Paso Salsa   Dip Tomato Mild 300g"
##   [7] "Smiths Crinkle Chips Salt & Vinegar 330g"
##   [8] "Grain Waves         Sweet Chilli 210g"
##   [9] "Doritos Corn Chip Mexican Jalapeno 150g"
##  [10] "Grain Waves Sour    Cream&Chives 210G"
##  [11] "Kettle Sensations   Siracha Lime 150g"
##  [12] "Twisties Cheese     270g"
##  [13] "WW Crinkle Cut      Chicken 175g"
##  [14] "Thins Chips Light&  Tangy 175g"
##  [15] "CCs Original 175g"
##  [16] "Burger Rings 220g"
##  [17] "NCC Sour Cream &    Garden Chives 175g"
##  [18] "Doritos Corn Chip Southern Chicken 150g"
##  [19] "Cheezels Cheese Box 125g"
##  [20] "Smiths Crinkle      Original 330g"
##  [21] "Infzns Crn Crnchers Tangy Gcamole 110g"
##  [22] "Kettle Sea Salt     And Vinegar 175g"
##  [23] "Smiths Chip Thinly  Cut Original 175g"
##  [24] "Kettle Original 175g"
##  [25] "Red Rock Deli Thai  Chilli&Lime 150g"
##  [26] "Pringles Sthrn FriedChicken 134g"
##  [27] "Pringles Sweet&Spcy BBQ 134g"
##  [28] "Red Rock Deli SR    Salsa & Mzzrlla 150g"
##  [29] "Thins Chips         Originl saltd 175g"
##  [30] "Red Rock Deli Sp    Salt & Truffle 150G"
##  [31] "Smiths Thinly       Swt Chli&S/Cream175G"
##  [32] "Kettle Chilli 175g"
##  [33] "Doritos Mexicana    170g"
##  [34] "Smiths Crinkle Cut  French OnionDip 150g"
##  [35] "Natural ChipCo      Hony Soy Chckn175g"
##  [36] "Dorito Corn Chp     Supreme 380g"
##  [37] "Twisties Chicken270g"
##  [38] "Smiths Thinly Cut   Roast Chicken 175g"
##  [39] "Smiths Crinkle Cut  Tomato Salsa 150g"
##  [40] "Kettle Mozzarella   Basil & Pesto 175g"
##  [41] "Infuzions Thai SweetChili PotatoMix 110g"
##  [42] "Kettle Sensations   Camembert & Fig 150g"
##  [43] "Smith Crinkle Cut   Mac N Cheese 150g"
##  [44] "Kettle Honey Soy    Chicken 175g"
##  [45] "Thins Chips Seasonedchicken 175g"
##  [46] "Smiths Crinkle Cut  Salt & Vinegar 170g"
##  [47] "Infuzions BBQ Rib   Prawn Crackers 110g"
##  [48] "GrnWves Plus Btroot & Chilli Jam 180g"
##  [49] "Tyrrells Crisps     Lightly Salted 165g"
##  [50] "Kettle Sweet Chilli And Sour Cream 175g"
##  [51] "Doritos Salsa       Medium 300g"
##  [52] "Kettle 135g Swt Pot Sea Salt"
##  [53] "Pringles SourCream  Onion 134g"
##  [54] "Doritos Corn Chips  Original 170g"
##  [55] "Twisties Cheese     Burger 250g"
##  [56] "Old El Paso Salsa   Dip Chnky Tom Ht300g"
##  [57] "Cobs Popd Swt/Chlli &Sr/Cream Chips 110g"
##  [58] "Woolworths Mild     Salsa 300g"
##  [59] "Natural Chip Co     Tmato Hrb&Spce 175g"
##  [60] "Smiths Crinkle Cut  Chips Original 170g"
##  [61] "Cobs Popd Sea Salt  Chips 110g"
##  [62] "Smiths Crinkle Cut  Chips Chs&Onion170g"
##  [63] "French Fries Potato Chips 175g"
##  [64] "Old El Paso Salsa   Dip Tomato Med 300g"
##  [65] "Doritos Corn Chips  Cheese Supreme 170g"
##  [66] "Pringles Original   Crisps 134g"
##  [67] "RRD Chilli&         Coconut 150g"
##  [68] "WW Original Corn    Chips 200g"
##  [69] "Thins Potato Chips  Hot & Spicy 175g"
##  [70] "Cobs Popd Sour Crm  &Chives Chips 110g"
##  [71] "Smiths Crnkle Chip  Orgnl Big Bag 380g"
##  [72] "Doritos Corn Chips  Nacho Cheese 170g"
##  [73] "Kettle Sensations   BBQ&Maple 150g"
##  [74] "WW D/Style Chip     Sea Salt 200g"
##  [75] "Pringles Chicken    Salt Crips 134g"
##  [76] "WW Original Stacked Chips 160g"
##  [77] "Smiths Chip Thinly  CutSalt/Vinegr175g"
##  [78] "Cheezels Cheese 330g"
##  [79] "Tostitos Lightly    Salted 175g"
##  [80] "Thins Chips Salt &  Vinegar 175g"
##  [81] "Smiths Crinkle Cut  Chips Barbecue 170g"
##  [82] "Cheetos Puffs 165g"
##  [83] "RRD Sweet Chilli &  Sour Cream 165g"
##  [84] "WW Crinkle Cut      Original 175g"
##  [85] "Tostitos Splash Of  Lime 175g"
##  [86] "Woolworths Medium   Salsa 300g"
##  [87] "Kettle Tortilla ChpsBtroot&Ricotta 150g"
##  [88] "CCs Tasty Cheese    175g"
##  [89] "Woolworths Cheese   Rings 190g"
##  [90] "Tostitos Smoked     Chipotle 175g"
##  [91] "Pringles Barbeque   134g"
##  [92] "WW Supreme Cheese   Corn Chips 200g"
##  [93] "Pringles Mystery    Flavour 134g"
##  [94] "Tyrrells Crisps     Ched & Chives 165g"
##  [95] "Snbts Whlgrn Crisps Cheddr&Mstrd 90g"
##  [96] "Cheetos Chs & Bacon Balls 190g"
##  [97] "Pringles Slt Vingar 134g"
##  [98] "Infuzions SourCream&Herbs Veg Strws 110g"
##  [99] "Kettle Tortilla ChpsFeta&Garlic 150g"
## [100] "Infuzions Mango     Chutny Papadums 70g"
## [101] "RRD Steak &         Chimuchurri 150g"
## [102] "RRD Honey Soy       Chicken 165g"
## [103] "Sunbites Whlegrn    Crisps Frch/Onin 90g"
## [104] "RRD Salt & Vinegar  165g"
## [105] "Doritos Cheese      Supreme 330g"
## [106] "Smiths Crinkle Cut  Snag&Sauce 150g"
## [107] "WW Sour Cream &OnionStacked Chips 160g"
## [108] "RRD Lime & Pepper   165g"
## [109] "Natural ChipCo Sea  Salt & Vinegr 175g"
## [110] "Red Rock Deli Chikn&Garlic Aioli 150g"
## [111] "RRD SR Slow Rst     Pork Belly 150g"
## [112] "RRD Pc Sea Salt     165g"
## [113] "Smith Crinkle Cut   Bolognese 150g"
## [114] "Doritos Salsa Mild  300g"
```

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.

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

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()`.

```r
#### Removing digits
productWords <-  productWords[grepl("\\d", words) == FALSE, ]
#### Removing special characters
productWords <-  productWords[grepl("[:alpha:]", words), ]
#### Let's look at the most common words by counting the number of times a word appears,
productWords[, .N, words][order(N, decreasing = TRUE)]
```

```
##              words  N
##   1:         Chips 21
##   2:        Smiths 16
##   3:       Crinkle 14
##   4:        Kettle 13
##   5:        Cheese 12
##  ---
## 127: Chikn&Garlic  1
## 128:         Aioli  1
## 129:         Slow  1
## 130:        Belly  1
## 131:    Bolognese  1
```

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

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

Next, we can use `summary()` to check summary statistics such as mean, min and max values for each feature to see if there are any obvious outliers in the data and if there are any nulls in any of the columns (`NA's : number of nulls` will appear in the output if there are any nulls).

```
summary(transactionData)
```

```
##       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  
```

```
sum(is.na(transactionData))
```

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

There are no nulls in the columns but product quantity appears to have an outlier which we should investigate further. Let's investigate further the case where 200 packets of chips are bought in one transaction.

```
#### Filter the dataset to find the outlier
transactionData %>% filter(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
```

There are two transactions where 200 packets of chips are bought in one transaction and both of these transactions were by the same customer.

```
#### Let's see if the customer has had other transactions
transactionData %>% filter(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
```

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
transactionData %>% filter(LYLTY_CARD_NBR != 226000) -> transactionData
#### Re-examine transaction data
summary(transactionData)
```

```
##       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
```

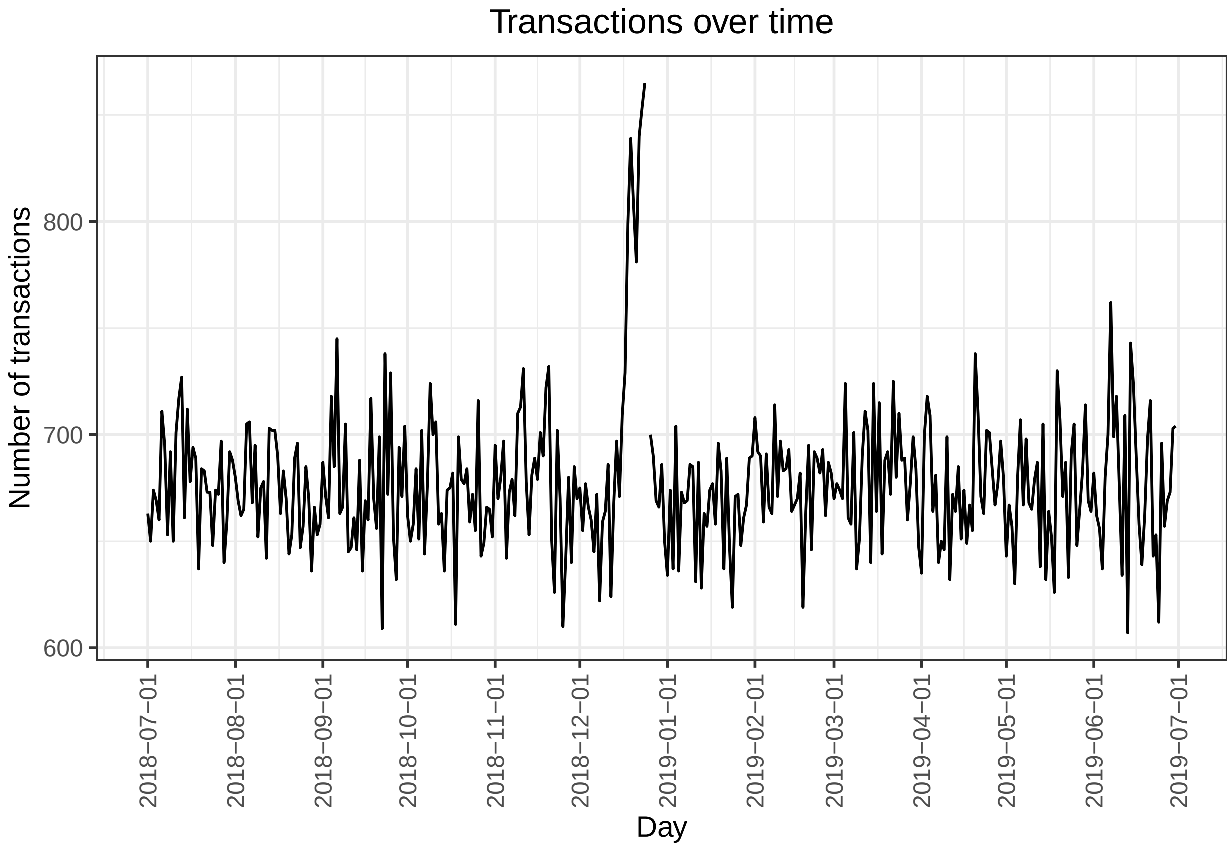
That's better. Now, let's look at the number of transaction lines over time to see if there are any obvious data issues such as missing data.

```
#### Count the number of transactions by date
transactionData %>% group_by(DATE) %>% summarise(number_of_trans_per_date=n()) ->
↪  NumOFtrans
NumOFtrans
```

```
## # A tibble: 364 x 2
##    DATE       number_of_trans_per_date
##    <date>                        <int>
##  1 2018-07-01                      663
##  2 2018-07-02                      650
##  3 2018-07-03                      674
##  4 2018-07-04                        669
##  5 2018-07-05                        660
##  6 2018-07-06                        711
##  7 2018-07-07                        695
##  8 2018-07-08                        653
##  9 2018-07-09                        692
## 10 2018-07-10                        650
## # ... with 354 more rows
```

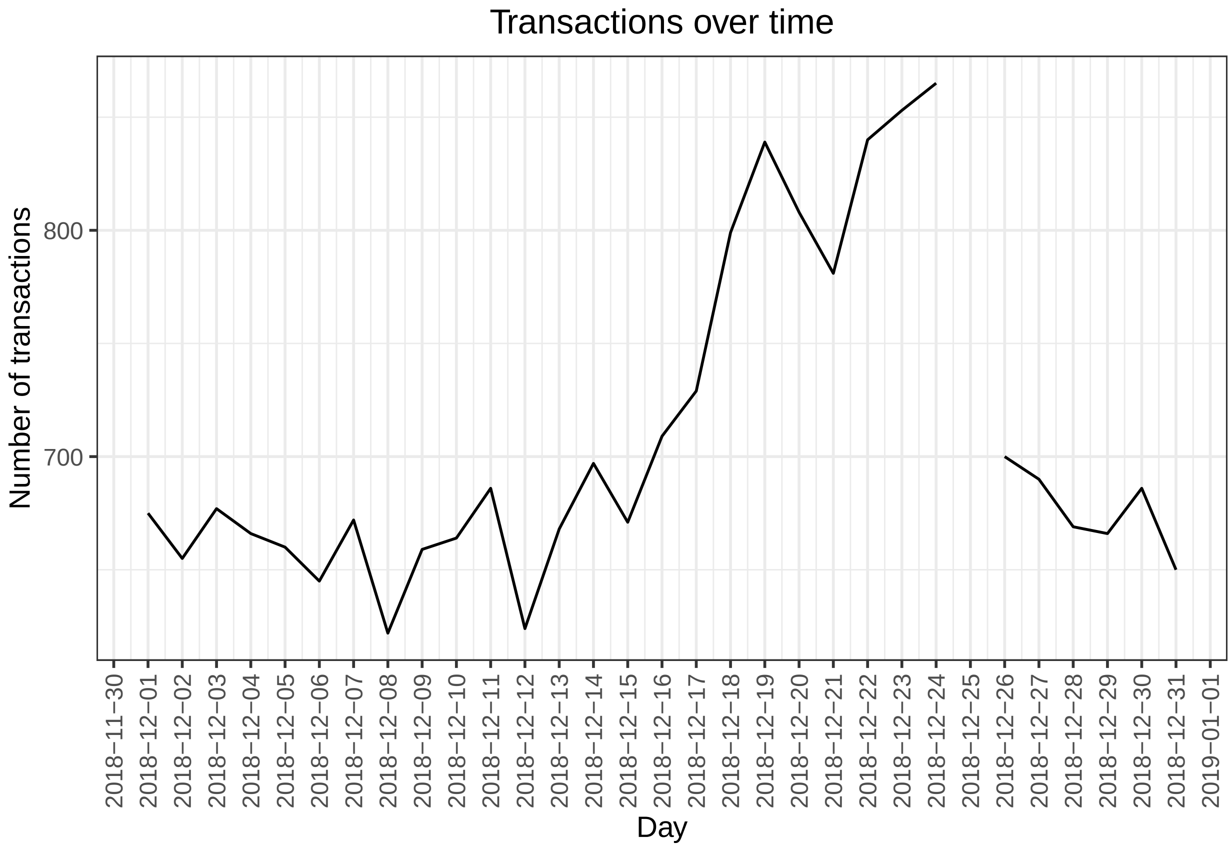
There's only 364 rows, meaning only 364 dates which indicates a missing date. Let's create a sequence of dates from 1 Jul 2018 to 30 Jun 2019 and use this to create a chart of number of transactions over time to find the missing date.

```
#### Creating a sequence of dates and join this the count of transactions by date
# creating a column of dates that includes every day from 1 Jul 2018 to 30 Jun 2019, and
↪  join it onto the data to fill in the missing day.
seqdates <-  data.table(seq(as.Date("2018/07/01"), as.Date("2019/06/30"), by ="day"))
setnames(seqdates,"DATE")
transactions_by_day <-  merge(seqdates, NumOFtrans,by="DATE", all.x = TRUE)
#### Setting plot themes to format graphs

theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
transactions_by_day$DATE <- as.Date(transactions_by_day$DATE)

#### Plot transactions over time
ggplot(transactions_by_day, aes(x = DATE, y = number_of_trans_per_date)) +
 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))
```



We can see that there is an increase in purchases in December and a break in late December. Let's zoom in on this.

```r
#### Filtering to December and look at individual days
# recreating the chart above zoomed in to the relevant dates.
ggplot(transactions_by_day[month(DATE) == 12, ], aes(x = DATE, y =
↪  number_of_trans_per_date)) +
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 the increase in sales occurs in the lead-up to Christmas and that there are zero sales on Christmas day itself. This is due to shops being closed on Christmas day. 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.

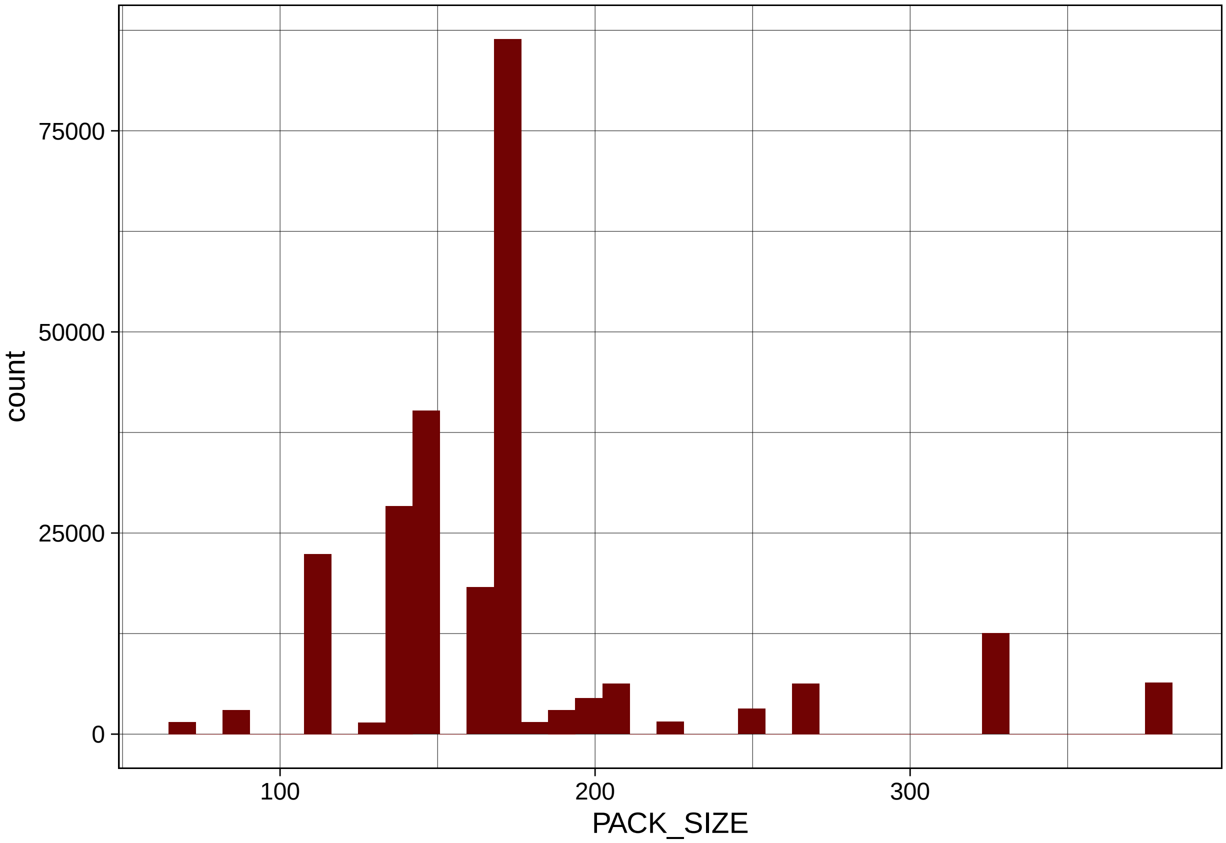
```
#### Pack size
#### We can work this out by taking the digits that are in PROD_NAME
transactionData %>% mutate(PACK_SIZE = parse_number(PROD_NAME)) -> transactionData
transactionData %>% group_by(PACK_SIZE) %>% summarise(number=n())
```

```
## # A tibble: 20 x 2
##    PACK_SIZE number
##        <dbl>  <int>
##  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
```

```
#### Let's check if the pack sizes look sensible
```

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

```
#### Let's plot a histogram of PACK_SIZE since we know that it is a categorical variable
↪  and not a continuous variable even though it is numeric.
ggplot(transactionData) +
  aes(x = PACK_SIZE) +
  geom_histogram(bins = 37L, fill = "#710303") +
  theme_linedraw()
```



```
# Over to you! Plot a histogram showing the number of transactions by pack size.
```

Pack sizes created look reasonable. Now to create brands, we can use the first word in PROD_NAME to work out the brand name…

```
#### Brands
# Creating a column which contains the brand of the product, by extracting it from the
↪  product name.
transactionData %>% mutate(BRAND = toupper(substr(PROD_NAME, 1, regexpr(pattern = ' ',
↪  PROD_NAME) - 1))) ->transactionData

transactionData%>% group_by(BRAND) %>% summarise(total=n())
```

```
## # A tibble: 28 x 2
##    BRAND    total
##    <chr>    <int>
##  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
## # ... with 18 more rows
```

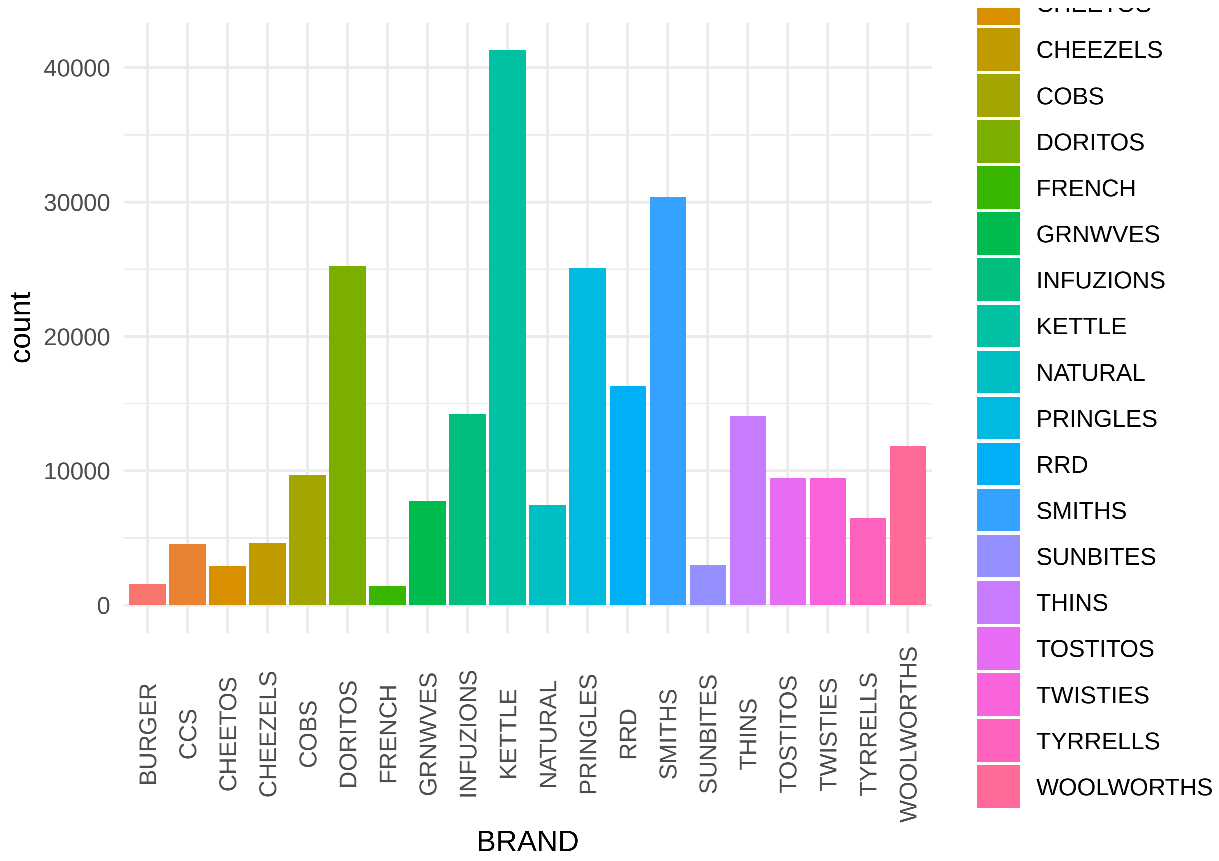
```
#### Checking brands
```

Some of the brand names look like they are of the same brands - such as RED and RRD, which are both Red Rock Deli chips. Let's combine these together.

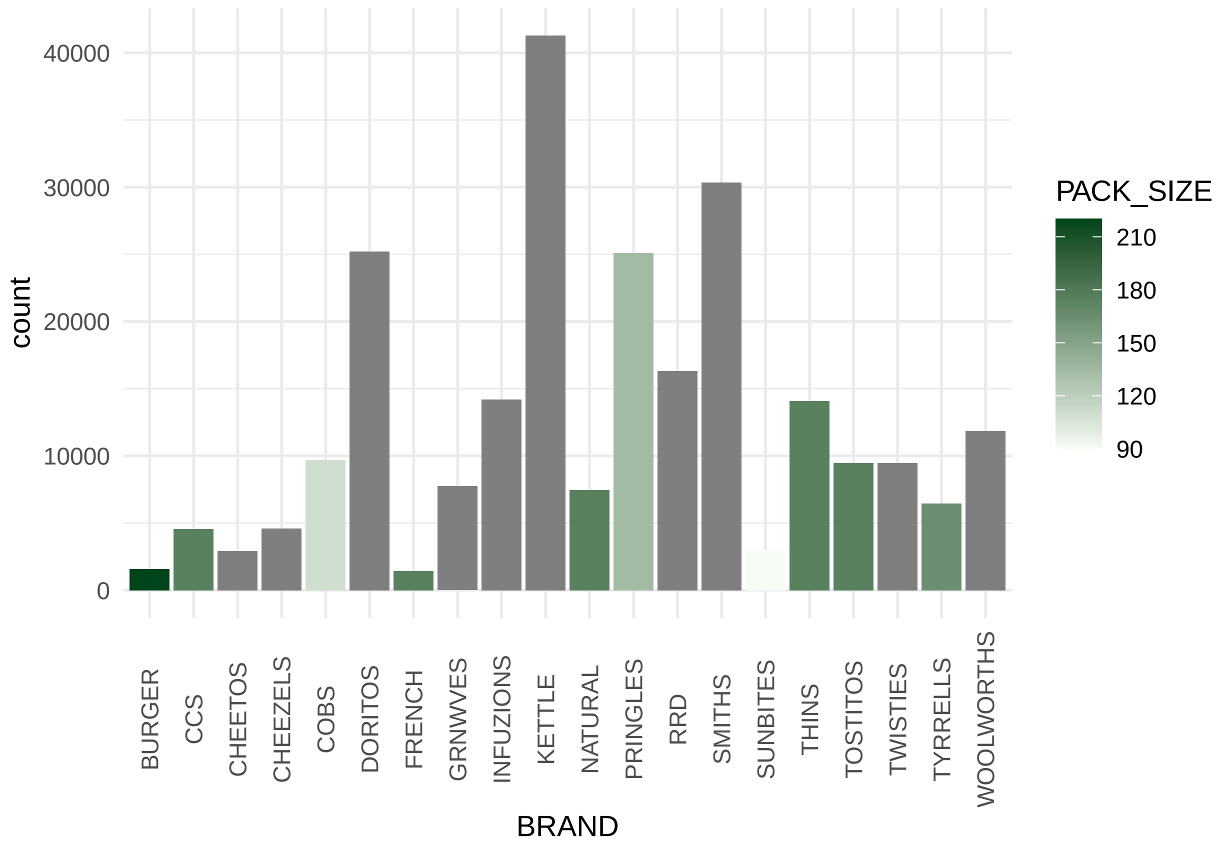
```
#### Clean brand names
transactionData[BRAND == "RED", BRAND := "RRD"]
transactionData[BRAND == "SNBTS", BRAND := "SUNBITES"]
transactionData[BRAND == "INFZNS", BRAND := "INFUZIONS"]
transactionData[BRAND == "WW", BRAND := "WOOLWORTHS"]
transactionData[BRAND == "SMITH", BRAND := "SMITHS"]
transactionData[BRAND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

transactionData%>% group_by(BRAND) %>% summarise(total=n())
```

```
## # A tibble: 20 x 2
##    BRAND      total
##    <chr>      <int>
##  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
```

```
ggplot(transactionData) +
  aes(x = BRAND, fill = BRAND) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal()+
        theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



```
ggplot(transactionData) +
  aes(x = BRAND, fill = PACK_SIZE) +
  geom_bar() +
  scale_fill_gradient(low = "#F7FCF5", high = "#00441B") +
  theme_minimal()+
        theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



Examining customer data Now that we are happy with the transaction dataset, let's have a look at the customer dataset.

```
#### Examining customer data.

summary(customerData)
```

```
##  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
```

```
sum(is.null(customerData))
```

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

```
customerData %>% group_by(LIFESTAGE) %>% summarise(total=n())
```

```
## # A tibble: 7 x 2
##   LIFESTAGE              total
##   <chr>                  <int>
## 1 MIDAGE SINGLES/COUPLES  7275
## 2 NEW FAMILIES            2549
## 3 OLDER FAMILIES          9780
## 4 OLDER SINGLES/COUPLES  14609
## 5 RETIREES               14805
## 6 YOUNG FAMILIES          9178
## 7 YOUNG SINGLES/COUPLES  14441
```

```
customerData %>% group_by(PREMIUM_CUSTOMER) %>% summarise(total=n())
```

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

```
#### Merge transaction data to customer data
fulldata <- merge(transactionData, customerData, all.x = TRUE)
```

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. Let's also check if some customers were not matched on by checking for nulls.

```
sum(is.null(fulldata$LIFESTAGE))
```

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

```
sum(is.null(fulldata$PREMIUM_CUSTOMER))
```

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

Great, there are no nulls! So all our customers in the transaction data has been accounted for in the customer dataset. Note that if you are continuing with Task 2, you may want to retain this dataset which you can write out as a csv

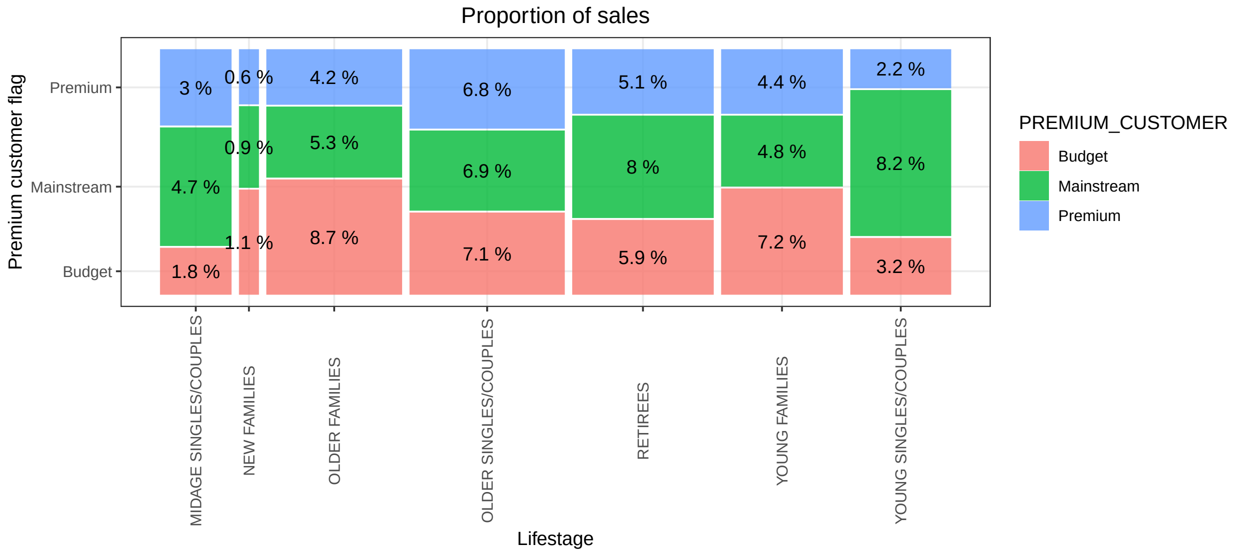
```
#fwrite(data, paste0(filePath,"QVI_data.csv"))
```

Data exploration is now complete!

Data analysis on customer segments Now that the data is ready for analysis, we can define some metrics of interest to the client:

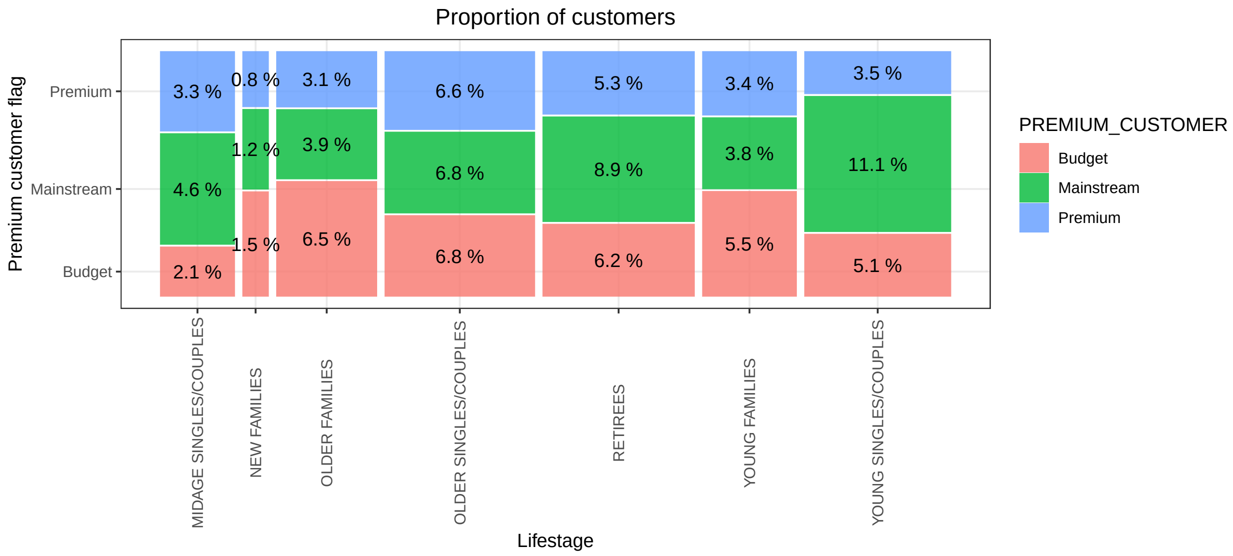
- Who spends the most on chips (total sales), describing customers by lifestage and how premium their general purchasing behaviour is
- How many customers are in each segment
- How many chips are bought per customer by segment
- What's the average chip price by customer segment We could also ask our 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.

```
#### Total sales by LIFESTAGE and PREMIUM_CUSTOMER
sales <-  fulldata[, .(SALES = sum(TOT_SALES)), .(LIFESTAGE, PREMIUM_CUSTOMER)]
p1 <- ggplot(data=sales) +
        geom_mosaic(aes( weight = SALES,x=product(PREMIUM_CUSTOMER,LIFESTAGE),fill =
        ↪   PREMIUM_CUSTOMER )) +
        labs(x = "Lifestage", y = "Premium customer flag", title = "Proportion of sales")
        ↪  +
        theme(axis.text.x = element_text(angle = 90, vjust = 0.5))

p1 + geom_text(data = ggplot_build(p1)$data[[1]], aes(x = (xmin + xmax)/2 , y =
 (ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)*100,'%'))))
```



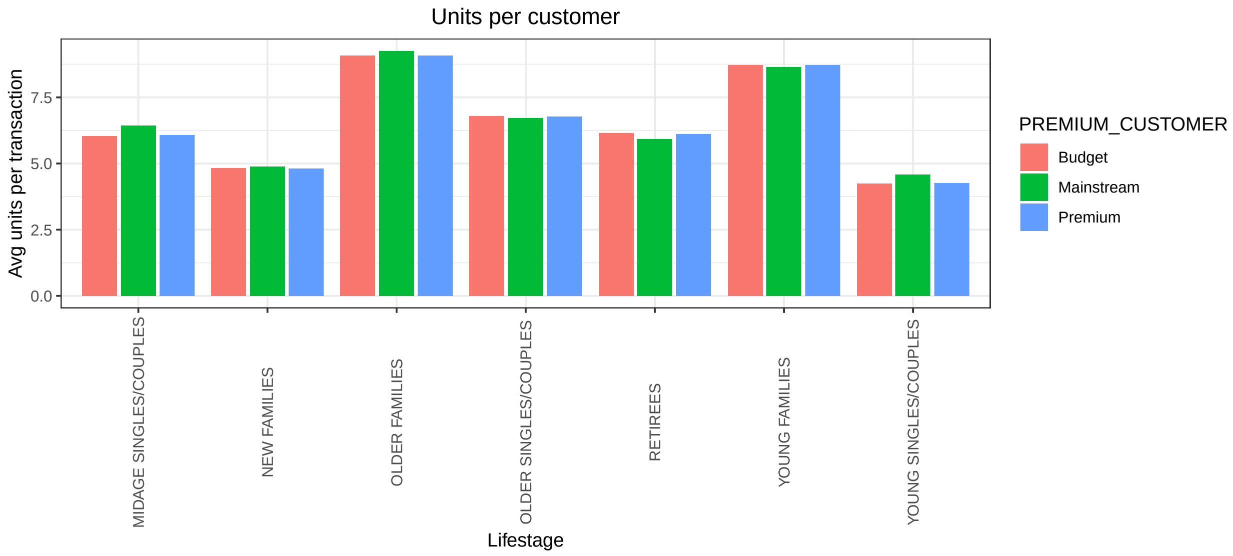
Sales are coming mainly from Budget - older families, Mainstream - young singles/couples, and Mainstream - retirees Let's see if the higher sales are due to there being more customers who buy chips.

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

p2 <- ggplot(data=customers) +
        geom_mosaic(aes( weight = CUSTOMERS,x=product(PREMIUM_CUSTOMER,LIFESTAGE),fill =
        ↪  PREMIUM_CUSTOMER )) +
        labs(x = "Lifestage", y = "Premium customer flag", title = "Proportion of
        ↪  customers") +
        theme(axis.text.x = element_text(angle = 90, vjust = 0.5))

p2 + geom_text(data = ggplot_build(p2)$data[[1]], aes(x = (xmin + xmax)/2 , y =
 (ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)*100,'%'))))
```



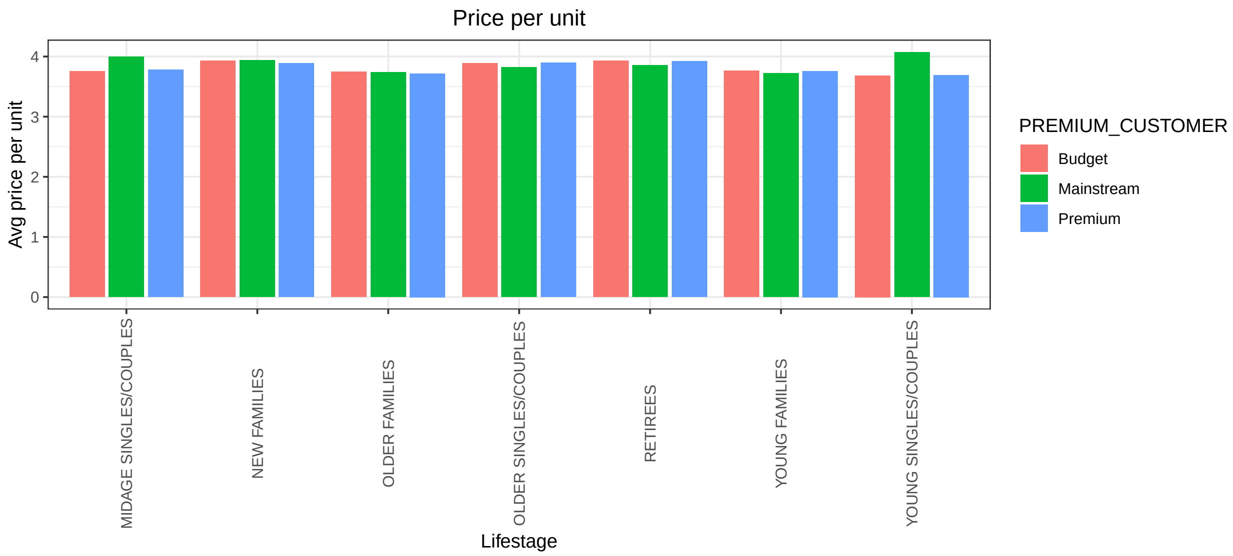
There are more Mainstream - young singles/couples and Mainstream - retirees who buy chips. This contributes to there being more sales to these customer segments but this is not a major driver for the Budget - Older families segment. Higher sales may also be driven by more units of chips being bought per customer. Let's have a look at this next.

```
#### Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER
avgunits <-
↪  fulldata[,.(AVG=sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR)),.(LIFESTAGE,PREMIUM_CUSTOMER)]

ggplot(data=avgunits,aes(weight=AVG,x=LIFESTAGE,fill=PREMIUM_CUSTOMER)) +
  geom_bar(position=position_dodge2())+
  labs(x="Lifestage",y="Avg units per transaction",title = "Units per customer") +
  theme(axis.text.x = element_text(angle=90,vjust=0.5))
```



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.

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

ggplot(data=avgprice,aes(weight=AVG,x=LIFESTAGE,fill=PREMIUM_CUSTOMER)) +
  geom_bar(position=position_dodge2())+
  labs(x="Lifestage",y="Avg price per unit",title = "Price per unit") +
  theme(axis.text.x = element_text(angle=90,vjust=0.5))
```



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. 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
fulldata %>% mutate(price= TOT_SALES/PROD_QTY) -> fulldata
t.test(fulldata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES")
 & PREMIUM_CUSTOMER == "Mainstream", price]
, fulldata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES")
 & PREMIUM_CUSTOMER != "Mainstream", price]
, alternative = "greater")
```

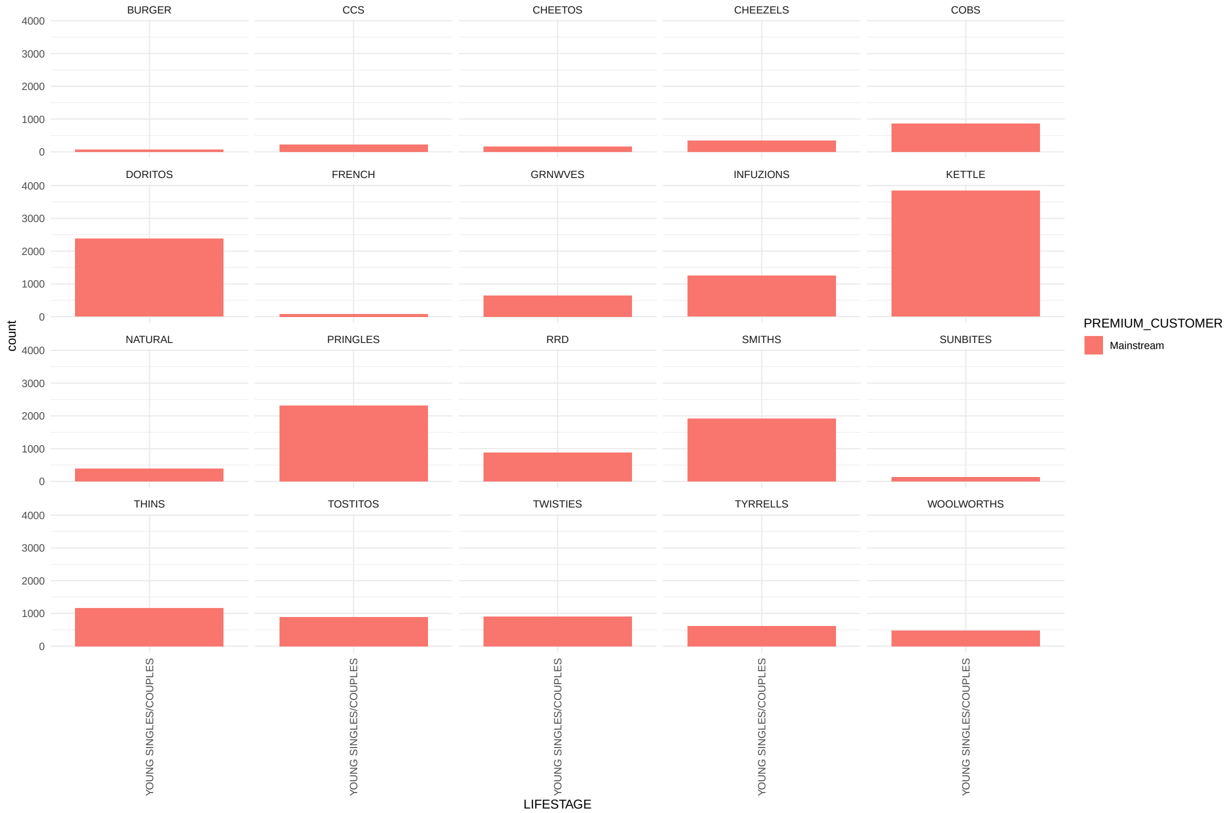
```
##
## Welch Two Sample t-test
##
## data: fulldata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE
SINGLES/COUPLES") & PREMIUM_CUSTOMER == "Mainstream", price] and
fulldata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") &
PREMIUM_CUSTOMER != "Mainstream", price]
## t = 37.624, 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
```

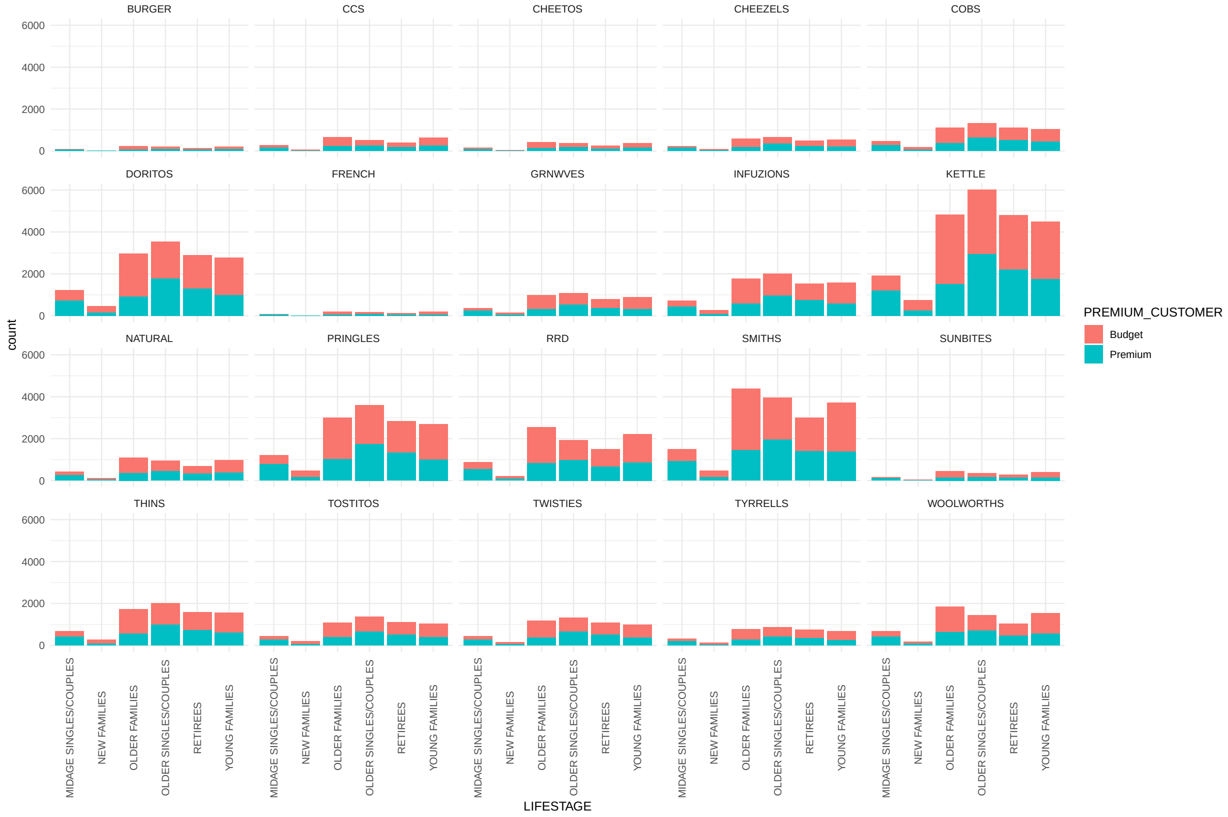
The t-test results in a p-value of 2.2e-16, i.e. the unit price for mainstream, young and mid-age singles and couples are significantly higher than that of budget or premium, young and midage singles and couples.

Deep dive into specific customer segments for insights

We have found quite a few interesting insights that we can dive deeper into. 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.

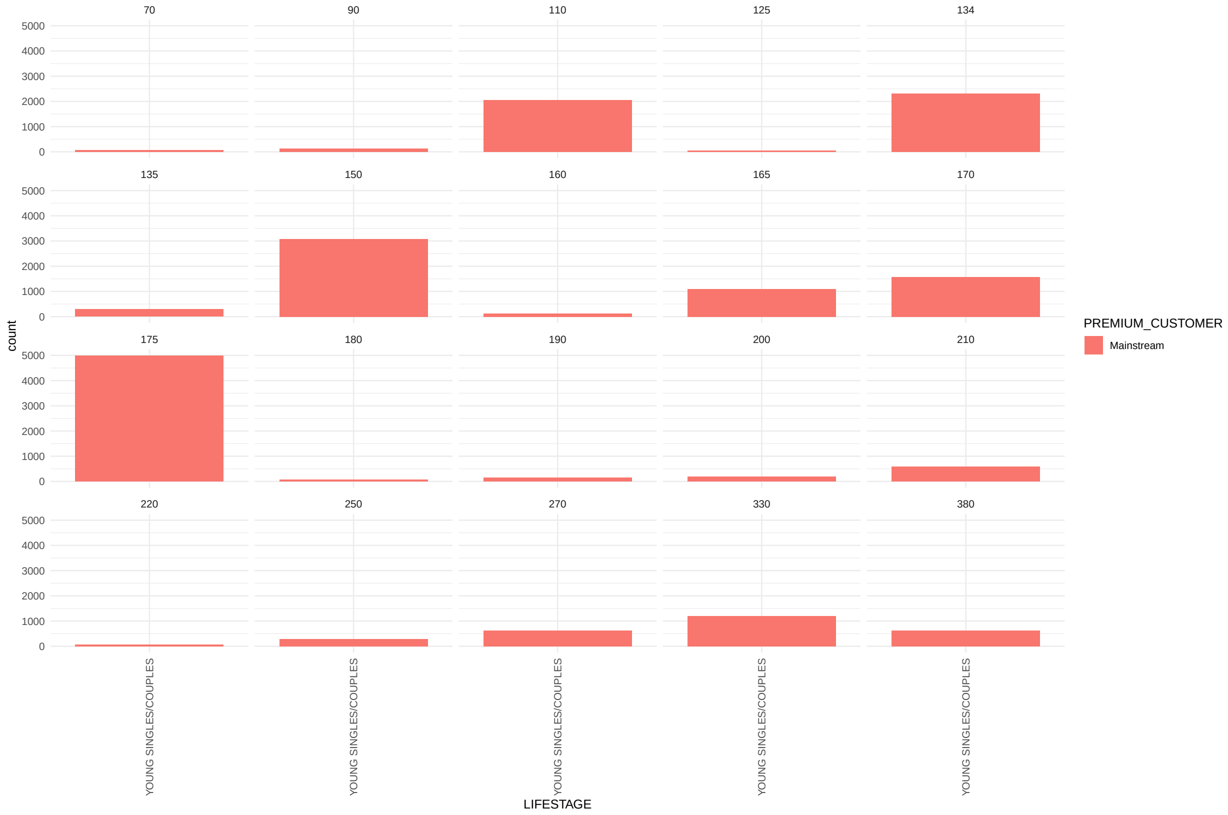
```
#### Deep dive into Mainstream, young singles/couples

a1 <- fulldata %>%
 filter(LIFESTAGE %in% "YOUNG SINGLES/COUPLES") %>%
 filter(PREMIUM_CUSTOMER %in% "Mainstream") %>%
 ggplot() +
  aes(x = LIFESTAGE, fill = PREMIUM_CUSTOMER) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal() +
  facet_wrap(vars(BRAND))+
  theme(axis.text.x = element_text(angle=90,vjust=0.5))

a2 <- fulldata %>%
 filter(!(LIFESTAGE %in% "YOUNG SINGLES/COUPLES")) %>%
 filter(!(PREMIUM_CUSTOMER %in% "Mainstream")) %>%
 ggplot() +
  aes(x = LIFESTAGE, fill = PREMIUM_CUSTOMER) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal() +
  facet_wrap(vars(BRAND))+
  theme(axis.text.x = element_text(angle=90,vjust=0.5))

library(patchwork)

a1 / a2
```

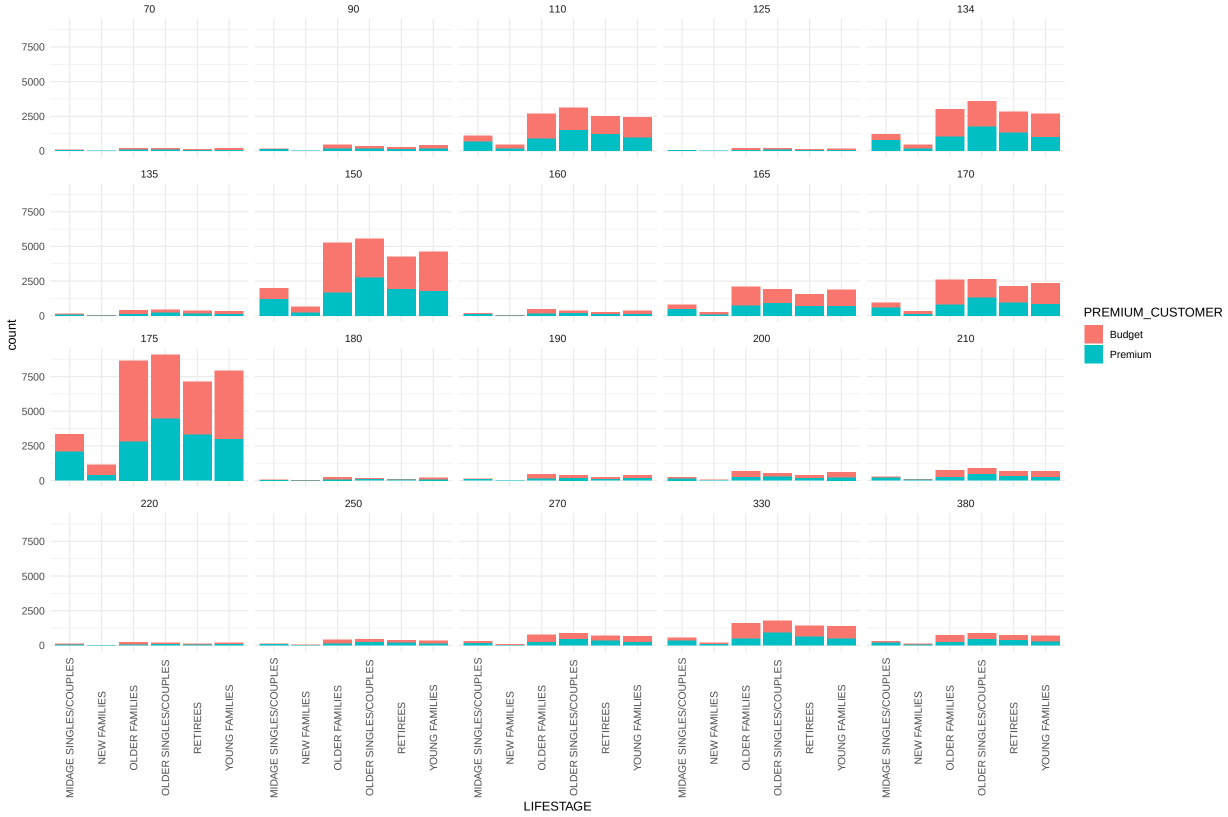




Let’s also find out if our target segment tends to buy larger packs of chips.

```r
#### Preferred pack size compared to the rest of the population
a4 <- fulldata %>%
 filter(!(LIFESTAGE %in% "YOUNG SINGLES/COUPLES")) %>%
 filter(!(PREMIUM_CUSTOMER %in% "Mainstream")) %>%
 ggplot() +
  aes(x = LIFESTAGE, fill = PREMIUM_CUSTOMER) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal() +
  facet_wrap(vars(PACK_SIZE)) +
  theme(axis.text.x = element_text(angle=90,vjust=0.5))


a3 <- fulldata %>%
 filter(LIFESTAGE %in% "YOUNG SINGLES/COUPLES") %>%
 filter(PREMIUM_CUSTOMER %in% "Mainstream") %>%
 ggplot() +
  aes(x = LIFESTAGE, fill = PREMIUM_CUSTOMER) +
  geom_bar() +
  scale_fill_hue(direction = 1) +
  theme_minimal() +
  facet_wrap(vars(PACK_SIZE)) +
  theme(axis.text.x = element_text(angle=90,vjust=0.5))

a3/a4
```


70
90
110
125
134
135
150
160
165
170
175
180
190
200
210
220
250
270
330
380
5000
4000
3000
2000
1000
0
count
YOUNG SINGLES/COUPLES
LIFESTAGE
PREMIUM_CUSTOMER
🟥 Mainstream


70
90
110
125
134
135
150
160
165
170
175
180
190
200
210
220
250
270
330
380
7500
5000
2500
0
count
MIDAGE SINGLES/COUPLES
NEW FAMILIES
OLDER FAMILIES
OLDER SINGLES/COUPLES
RETIREES
YOUNG FAMILIES
LIFESTAGE
PREMIUM_CUSTOMER
🟥 Budget
🟦 Premium