Quanitum_virtual_internship_1

Shrinath Rajeshirke

26/02/2022

```
### Load required libraries
library(data.table)
library(ggplot2)
library(ggmosaic)
library(readr)
library(stringr)
library(dplyr)

##
## 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

Transactiondata <-
fread(paste0("C:/Users/ASUS/Desktop/Quantium/QVI_transaction_data.csv"))
Customerdata <-
fread(paste0("C:/Users/ASUS/Desktop/Quantium/QVI_purchase_behaviour.csv"))
```

Exploratory Data Analysis

```
head(Transactiondata)

##     DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 1: 43390         1           1000      1        5
## 2: 43599         1           1307    348       66
## 3: 43605         1           1343    383       61
## 4: 43329         2           2373    974       69
## 5: 43330         2           2426   1038      108
## 6: 43604         4           4074   2982       57
##                                 PROD_NAME PROD_QTY TOT_SALES
## 1:   Natural Chip        Compny SeaSalt175g        2       6.0
## 2:                 CCs Nacho Cheese    175g        3       6.3
## 3:   Smiths Crinkle Cut  Chips Chicken 170g        2       2.9
## 4:   Smiths Chip Thinly  S/Cream&Onion 175g        5      15.0
```

```
## 5: Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13.8
## 6: Old El Paso Salsa   Dip Tomato Mild 300g        1       5.1

str(Transactiondata)

## Classes 'data.table' and 'data.frame':   264836 obs. of  8 variables:
##  $ DATE          : int  43390 43599 43605 43329 43330 43604 43601 43601
43332 43330 ...
##  $ STORE_NBR     : int  1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: int  1000 1307 1343 2373 2426 4074 4149 4196 5026 7150
...
##  $ TXN_ID        : int  1 348 383 974 1038 2982 3333 3539 4525 6900 ...
##  $ PROD_NBR      : int  5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr  "Natural Chip        Compny SeaSalt175g" "CCs
Nacho Cheese    175g" "Smiths Crinkle Cut  Chips Chicken 170g" "Smiths Chip
Thinly  S/Cream&Onion 175g" ...
##  $ PROD_QTY      : int  2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num  6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

We can see that the date column is in an integer format. we have to change it to date format.

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

table(Transactiondata$PROD_NAME)

##
##                        Burger Rings 220g
##                                     1564
##                CCs Nacho Cheese    175g
##                                     1498
##                       CCs Original 175g
##                                     1514
##                CCs Tasty Cheese    175g
##                                     1539
##           Cheetos Chs & Bacon Balls 190g
##                                     1479
##                      Cheetos Puffs 165g
##                                     1448
##                     Cheezels Cheese 330g
##                                     3149
##                Cheezels Cheese Box 125g
##                                     1454
##           Cobs Popd Sea Salt  Chips 110g
##                                     3265
##   Cobs Popd Sour Crm  &Chives Chips 110g
##                                     3159
## Cobs Popd Swt/Chlli &Sr/Cream Chips 110g
##                                     3269
```

```
##         Dorito Corn Chp     Supreme 380g
##                                     3185
##         Doritos Cheese      Supreme 330g
##                                     3052
##  Doritos Corn Chip Mexican Jalapeno 150g
##                                     3204
##  Doritos Corn Chip Southern Chicken 150g
##                                     3172
##  Doritos Corn Chips  Cheese Supreme 170g
##                                     3217
##    Doritos Corn Chips  Nacho Cheese 170g
##                                     3160
##        Doritos Corn Chips  Original 170g
##                                     3121
##                Doritos Mexicana    170g
##                                     3115
##          Doritos Salsa       Medium 300g
##                                     1449
##                Doritos Salsa Mild  300g
##                                     1472
##           French Fries Potato Chips 175g
##                                     1418
##    Grain Waves         Sweet Chilli 210g
##                                     3167
##    Grain Waves Sour    Cream&Chives 210G
##                                     3105
##    GrnWves Plus Btroot & Chilli Jam 180g
##                                     1468
##  Infuzions BBQ Rib   Prawn Crackers 110g
##                                     3174
##  Infuzions Mango     Chutny Papadums 70g
##                                     1507
## Infuzions SourCream&Herbs Veg Strws 110g
##                                     3134
## Infuzions Thai SweetChili PotatoMix 110g
##                                     3242
##   Infzns Crn Crnchers Tangy Gcamole 110g
##                                     3144
##             Kettle 135g Swt Pot Sea Salt
##                                     3257
##                       Kettle Chilli 175g
##                                     3038
##         Kettle Honey Soy    Chicken 175g
##                                     3148
##   Kettle Mozzarella   Basil & Pesto 175g
##                                     3304
##                     Kettle Original 175g
##                                     3159
##     Kettle Sea Salt     And Vinegar 175g
##                                     3173
```

```
##       Kettle Sensations   BBQ&Maple 150g
##                                     3083
## Kettle Sensations   Camembert & Fig 150g
##                                     3219
##    Kettle Sensations   Siracha Lime 150g
##                                     3127
##  Kettle Sweet Chilli And Sour Cream 175g
##                                     3200
##  Kettle Tortilla ChpsBtroot&Ricotta 150g
##                                     3146
##     Kettle Tortilla ChpsFeta&Garlic 150g
##                                     3138
## Kettle Tortilla ChpsHny&Jlpno Chili 150g
##                                     3296
##   Natural Chip        Compny SeaSalt175g
##                                     1468
##  Natural Chip Co     Tmato Hrb&Spce 175g
##                                     1572
##   Natural ChipCo      Hony Soy Chckn175g
##                                     1460
##   Natural ChipCo Sea  Salt & Vinegr 175g
##                                     1550
##   NCC Sour Cream &    Garden Chives 175g
##                                     1419
## Old El Paso Salsa   Dip Chnky Tom Ht300g
##                                     3125
##  Old El Paso Salsa   Dip Tomato Med 300g
##                                     3114
## Old El Paso Salsa   Dip Tomato Mild 300g
##                                     3085
##                 Pringles Barbeque   134g
##                                     3210
##      Pringles Chicken    Salt Crips 134g
##                                     3104
##         Pringles Mystery    Flavour 134g
##                                     3114
##          Pringles Original   Crisps 134g
##                                     3157
##                 Pringles Slt Vingar 134g
##                                     3095
##           Pringles SourCream  Onion 134g
##                                     3162
##         Pringles Sthrn FriedChicken 134g
##                                     3083
##             Pringles Sweet&Spcy BBQ 134g
##                                     3177
##    Red Rock Deli Chikn&Garlic Aioli 150g
##                                     1434
##  Red Rock Deli Sp    Salt & Truffle 150G
##                                     1498
```

```
## Red Rock Deli SR    Salsa & Mzzrlla 150g
##                                     1458
##     Red Rock Deli Thai  Chilli&Lime 150g
##                                     1495
##         RRD Chilli&         Coconut 150g
##                                     1506
##         RRD Honey Soy       Chicken 165g
##                                     1513
##                 RRD Lime & Pepper   165g
##                                     1473
##                 RRD Pc Sea Salt     165g
##                                     1431
##                 RRD Salt & Vinegar  165g
##                                     1474
##      RRD SR Slow Rst     Pork Belly 150g
##                                     1526
##     RRD Steak &         Chimuchurri 150g
##                                     1455
##      RRD Sweet Chilli &  Sour Cream 165g
##                                     1516
##       Smith Crinkle Cut   Bolognese 150g
##                                     1451
##    Smith Crinkle Cut   Mac N Cheese 150g
##                                     1512
##    Smiths Chip Thinly  Cut Original 175g
##                                     1614
##   Smiths Chip Thinly  CutSalt/Vinegr175g
##                                     1440
##   Smiths Chip Thinly  S/Cream&Onion 175g
##                                     1473
##        Smiths Crinkle      Original 330g
##                                     3142
## Smiths Crinkle Chips Salt & Vinegar 330g
##                                     3197
##  Smiths Crinkle Cut  Chips Barbecue 170g
##                                     1489
##   Smiths Crinkle Cut  Chips Chicken 170g
##                                     1484
##  Smiths Crinkle Cut  Chips Chs&Onion170g
##                                     1481
##  Smiths Crinkle Cut  Chips Original 170g
##                                     1461
## Smiths Crinkle Cut  French OnionDip 150g
##                                     1438
##  Smiths Crinkle Cut  Salt & Vinegar 170g
##                                     1455
##      Smiths Crinkle Cut  Snag&Sauce 150g
##                                     1503
##    Smiths Crinkle Cut  Tomato Salsa 150g
##                                     1470
```

```
##   Smiths Crnkle Chip  Orgnl Big Bag 380g
##                                     3233
## Smiths Thinly       Swt Chli&S/Cream175G
##                                     1461
##   Smiths Thinly Cut   Roast Chicken 175g
##                                     1519
##     Snbts Whlgrn Crisps Cheddr&Mstrd 90g
##                                     1576
## Sunbites Whlegrn    Crisps Frch/Onin 90g
##                                     1432
##   Thins Chips         Originl saltd 175g
##                                     1441
##           Thins Chips Light&  Tangy 175g
##                                     3188
##         Thins Chips Salt &  Vinegar 175g
##                                     3103
##         Thins Chips Seasonedchicken 175g
##                                     3114
##     Thins Potato Chips  Hot & Spicy 175g
##                                     3229
##          Tostitos Lightly    Salted 175g
##                                     3074
##        Tostitos Smoked     Chipotle 175g
##                                     3145
##           Tostitos Splash Of  Lime 175g
##                                     3252
##                Twisties Cheese     270g
##                                     3115
##          Twisties Cheese     Burger 250g
##                                     3169
##                      Twisties Chicken270g
##                                     3170
##   Tyrrells Crisps     Ched & Chives 165g
##                                     3268
##  Tyrrells Crisps     Lightly Salted 165g
##                                     3174
##           Woolworths Cheese   Rings 190g
##                                     1516
##           Woolworths Medium   Salsa 300g
##                                     1430
##           Woolworths Mild     Salsa 300g
##                                     1491
##         WW Crinkle Cut      Chicken 175g
##                                     1467
##        WW Crinkle Cut      Original 175g
##                                     1410
##        WW D/Style Chip     Sea Salt 200g
##                                     1469
##           WW Original Corn    Chips 200g
##                                     1495
```

```
##           WW Original Stacked Chips 160g
##                                       1487
##   WW Sour Cream &OnionStacked Chips 160g
##                                       1483
##      WW Supreme Cheese   Corn Chips 200g
##                                       1509
```

These are the different types of product names. We are looking at potato chips. Let's do some basic text analysis by summarising the individual words in the product name.

```
#### Examine the words in PROD_NAME to see if there are any incorrect entries
such
#### as products that are not chips
productWords <- data.table(unlist(strsplit(unique(Transactiondata[,
PROD_NAME]), " ")))
setnames(productWords, 'words')
```

We are only interested in words that will tell us if the product is chips or not, we remove all words with digits and special characters such as '&' from our set of product words.

```
#### Removing digits
productWords[, digit := grepl("[0-9]", words)]
productWords1 <- productWords[digit == FALSE, ][,digit := NULL]
#### Removing special characters
productWords1[, spchr := grepl("[&/]", words)]
productWords2 <- productWords1[spchr == FALSE, ][,spchr := NULL]
#### Let's look at the most common words by counting the number of times a
word appears and
#### sorting them by this frequency in order of highest to lowest frequency
sort(table(productWords2),decreasing = TRUE)
```

```
## productWords2
##                          Chips          Smiths         Crinkle
Cut
##            234             21             16              14
14
##         Kettle         Cheese           Salt        Original
Chip
##             13             12             12              10
9
##        Doritos          Salsa           Corn        Pringles
RRD
##              9              9              8               8
8
##        Chicken             WW            Sea            Sour
Chilli
##              7              7              6               6
5
##         Crisps         Thinly          Thins         Vinegar
Cream
##              5              5              5               5
```

```
4
##            Deli       Infuzions         Natural             Red
Rock
##               4               4               4               4
4
##         Supreme             CCs            Cobs             Dip
El
##               4               3               3               3
3
##            Lime            Mild             Old            Paso
Popd
##               3               3               3               3
3
##      Sensations             Soy           Sweet          Tomato
Tortilla
##               3               3               3               3
3
##        Tostitos        Twisties      Woolworths             And
BBQ
##               3               3               3               2
2
##          Burger         Cheetos        Cheezels          ChipCo
Chives
##               2               2               2               2
2
##          French           Grain           Honey         Lightly
Medium
##               2               2               2               2
2
##           Nacho          Potato           Rings          Salted
Smith
##               2               2               2               2
2
##              SR             Swt           Tangy            Thai
Tyrrells
##               2               2               2               2
2
##           Waves           Aioli           Bacon             Bag
Balls
##               2               1               1               1
1
##        Barbecue        Barbeque           Basil           Belly
Big
##               1               1               1               1
1
##       Bolognese             Box          Btroot       Camembert
Ched
##               1               1               1               1
1
##           Chili     Chimuchurri        Chipotle            Chnky
```

```
Chp
##               1               1               1               1
1
##             Chs          Chutny              Co         Coconut
Compny
##               1               1               1               1
1
##        Crackers           Crips             Crm             Crn
Crnchers
##               1               1               1               1
1
##          Crnkle          Dorito             Fig         Flavour
FriedChicken
##               1               1               1               1
1
##           Fries          Garden         Gcamole         GrnWves
Hony
##               1               1               1               1
1
##             Hot          Infzns        Jalapeno             Jam
Mac
##               1               1               1               1
1
##           Mango             Med         Mexican        Mexicana
Mozzarella
##               1               1               1               1
1
##         Mystery         Mzzrlla               N             NCC
Of
##               1               1               1               1
1
##           Onion        OnionDip           Orgnl         Originl
Papadums
##               1               1               1               1
1
##              Pc          Pepper           Pesto            Plus
Pork
##               1               1               1               1
1
##             Pot       PotatoMix           Prawn           Puffs
Rib
##               1               1               1               1
1
##           Roast             Rst           saltd Seasonedchicken
Siracha
##               1               1               1               1
1
##            Slow             Slt          Smoked           Snbts
SourCream
##               1               1               1               1
```

```
1
##        Southern              Sp           Spicy          Splash
Stacked
##               1               1               1               1
1
##           Steak           Sthrn           Strws        Sunbites
SweetChili
##               1               1               1               1
1
##           Tasty           Tmato             Tom         Truffle
Veg
##               1               1               1               1
1
##          Vinegr          Vingar         Whlegrn          Whlgrn
##               1               1               1               1
```

There are salsa products in the dataset but we are only interested in the chips category remove these.

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

#### summary of dataset for checking null values and outliers
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
```

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

```
#### Filter the dataset to find the outlier
count_200<- Transactiondata[Transactiondata$PROD_QTY==200]
count_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
customer <- Transactiondata[Transactiondata$LYLTY_CARD_NBR == 226000]
customer
```

```
##          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 only two transactions done by this customer.

Removing customer from the dataset

```
Transactiondata <- Transactiondata[!(Transactiondata$LYLTY_CARD_NBR ==
226000)]
```

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
countByDate <- count(Transactiondata,Transactiondata$DATE)
### number of rows
nrow(countByDate)
```

```
## [1] 364
```

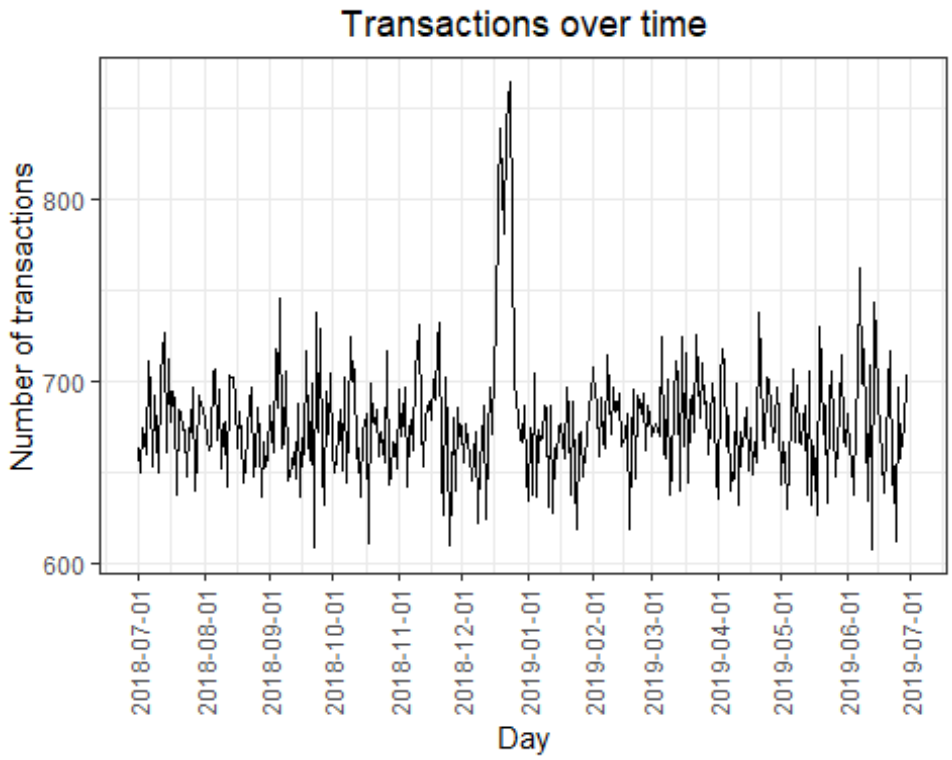
```
### summary of transactions
summary(countByDate)
```

```
##  Transactiondata$DATE       n
##  Min.   :2018-07-01   Min.   :607.0
##  1st Qu.:2018-09-29   1st Qu.:658.0
##  Median :2018-12-30   Median :674.0
##  Mean   :2018-12-30   Mean   :677.9
##  3rd Qu.:2019-03-31   3rd Qu.:694.2
##  Max.   :2019-06-30   Max.   :865.0
```

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.

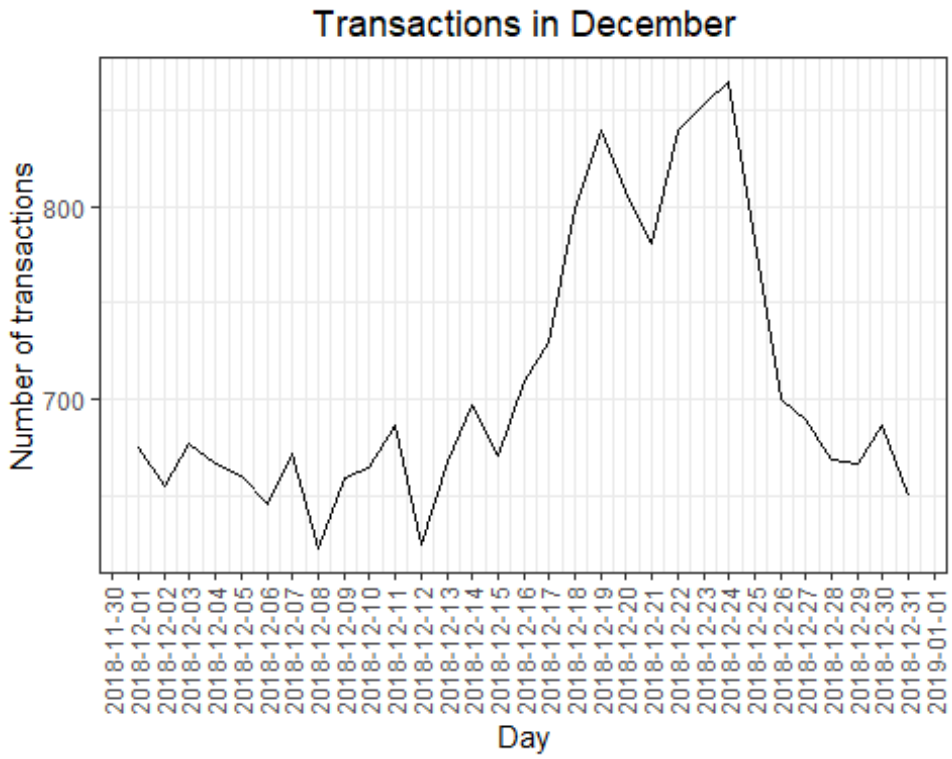
```
### create a sequence of dates and join this count of transaction by date
transaction_by_day <- Transactiondata[order(DATE),]
```

```
#### Setting plot themes to format graphs
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))
transOverTime <-ggplot(countByDate, aes(x =
countByDate$`Transactiondata$DATE`, y = countByDate$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))
transOverTime
```



We can see that there is an increase in purchases in December and a break in late December. Lets recreate chart only for december month to see in depth.

```
filterData <- countByDate[countByDate$`Transactiondata$DATE` >= "2018-12-01"
& countByDate$`Transactiondata$DATE` <= "2018-12-31"]

ggplot(filterData, aes(x = filterData$`Transactiondata$DATE`, y =
filterData$n)) +
 geom_line() +
 labs(x = "Day", y = "Number of transactions", title = "Transactions in
December") +
 scale_x_date(breaks = "1 day") +
 theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



We can see in the plot, the increase in sales occurs in the start of Christmas. There are no sales on christmas day itself. This is due to shops are closed on christmas day. Since, there are no outliers. we can move on to creating other features such as brand of chips or pack size from PROD_NAME.

```
#### Pack size
#### We can work this out by taking the digits that are in PROD_NAME
Transactiondata[, PACK_SIZE := parse_number(PROD_NAME)]
#### check output
#### Let's check if the pack sizes look sensible
Transactiondata[, .N, PACK_SIZE][order(PACK_SIZE)]
```

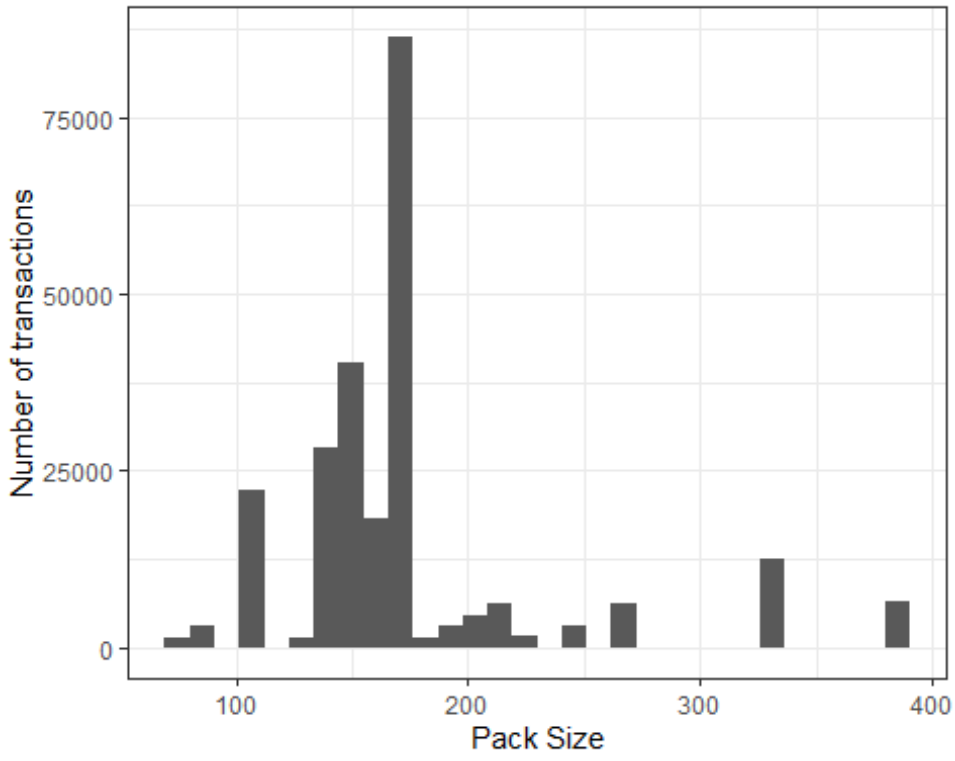
```
##     PACK_SIZE     N
##  1:        70  1507
##  2:        90  3008
##  3:       110 22387
##  4:       125  1454
##  5:       134 25102
##  6:       135  3257
##  7:       150 40203
##  8:       160  2970
##  9:       165 15297
## 10:       170 19983
## 11:       175 66390
## 12:       180  1468
## 13:       190  2995
## 14:       200  4473
```

```
## 15:       210  6272
## 16:       220  1564
## 17:       250  3169
## 18:       270  6285
## 19:       330 12540
## 20:       380  6416
```

The largest size is 380g and the smallest size is 70g.

```
#### Let's plot a histogram of PACK_SIZE
ggplot(Transactiondata, aes(x=PACK_SIZE))+
  geom_histogram()+
  xlab("Pack Size")+
  ylab("Number of transactions")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



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
# Over to you! Create a column which contains the brand of the product,
Transactiondata[, BRAND := str_extract(PROD_NAME, "\\w+")]
#### check output
#### Let's check if the pack sizes look sensible
Transactiondata[, .N, BRAND][order(BRAND)]
```

```
##          BRAND     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     N
```

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 == "Doritos", BRAND := "Doritos"]
Transactiondata[BRAND == "Grain", BRAND := "GrnWves"]
Transactiondata[BRAND == "Infuzions", BRAND := "Infzns"]
Transactiondata[BRAND == "Natural", BRAND := "NCC"]
Transactiondata[BRAND == "Smiths", BRAND := "Smith"]
Transactiondata[BRAND == "Sunbites", BRAND := "Snbts"]
Transactiondata[BRAND == "Woodworths", BRAND := "WW"]

#### Check again
Transactiondata[, .N, by = BRAND][order(BRAND)]
```

```
##          BRAND     N
##  1:     Burger  1564
##  2:        CCs  4551
```

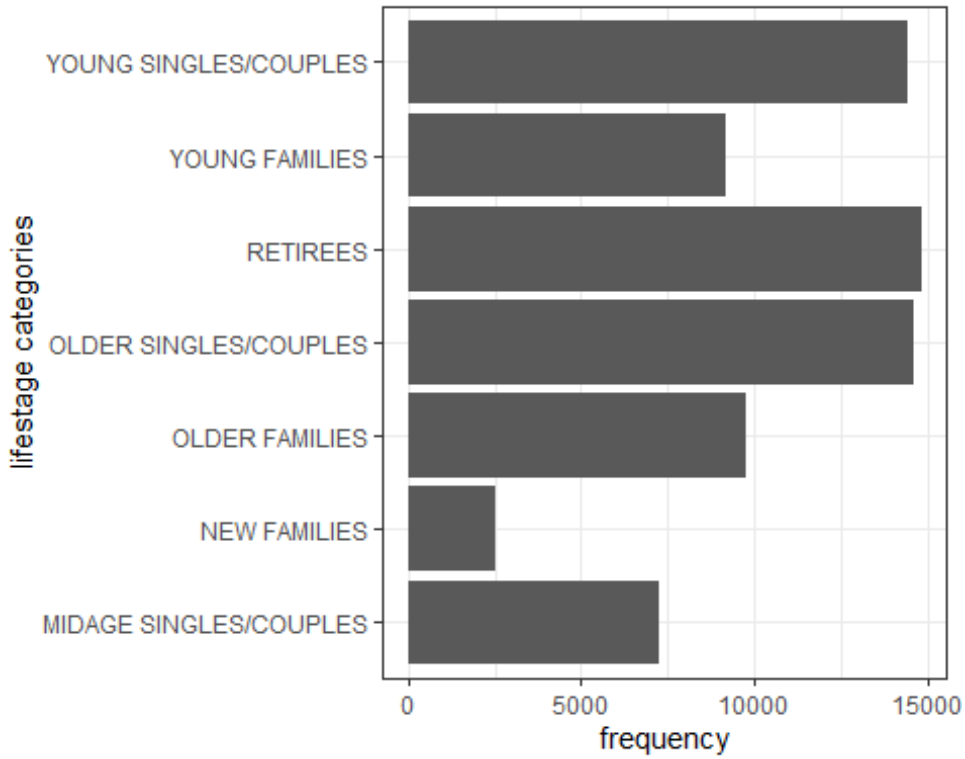
```
##  3:    Cheetos  2927
##  4:   Cheezels  4603
##  5:       Cobs  9693
##  6:     Dorito  3183
##  7:    Doritos 22041
##  8:     French  1418
##  9:    GrnWves  7740
## 10:     Infzns 14201
## 11:     Kettle 41288
## 12:        NCC  7469
## 13:   Pringles 25102
## 14:        RRD 11894
## 15:        Red  4427
## 16:      Smith 30353
## 17:      Snbts  3008
## 18:      Thins 14075
## 19:   Tostitos  9471
## 20:   Twisties  9454
## 21:   Tyrrells  6442
## 22:         WW 10320
## 23: Woolworths  1516
##          BRAND     N
```

Examining customer data

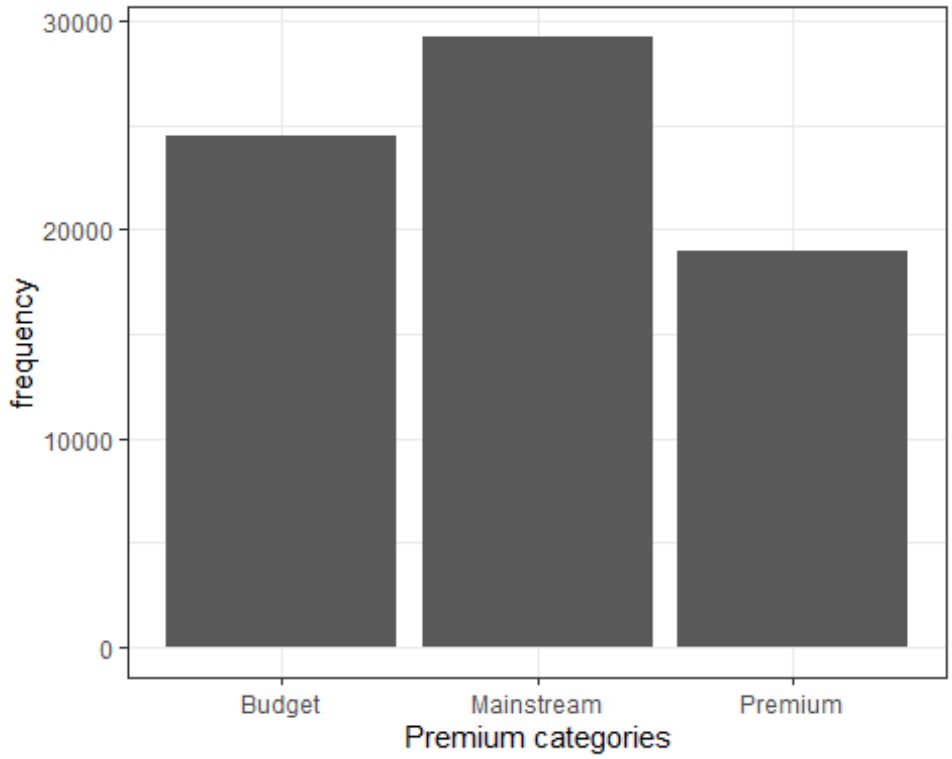
Now that we are happy with the transaction dataset, let's have a look at the customer dataset.

```
#### 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

### categories of LIFESTAGE
unique(Customerdata$LIFESTAGE)

## [1] "YOUNG SINGLES/COUPLES"  "YOUNG FAMILIES"         "OLDER
SINGLES/COUPLES"
## [4] "MIDAGE SINGLES/COUPLES" "NEW FAMILIES"           "OLDER FAMILIES"
## [7] "RETIREES"

### CAtegories of PREMIUM_CUSTOMER
unique(Customerdata$PREMIUM_CUSTOMER)

## [1] "Premium"    "Mainstream" "Budget"
```

```r
### plot of LIFESTAGE
ggplot(Customerdata, aes(x=LIFESTAGE))+
  geom_bar()+
  xlab("lifestage categories")+
  ylab("frequency")+
  coord_flip()
```




```r
### Plot of premium customer
ggplot(Customerdata, aes(x=PREMIUM_CUSTOMER))+
  geom_bar()+
  xlab("Premium categories")+
  ylab("frequency")
```




```
#### Merge transaction data to customer data
data <- merge(Transactiondata, Customerdata, all.x = TRUE)
```

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

```
sum(is.na(data))
```

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

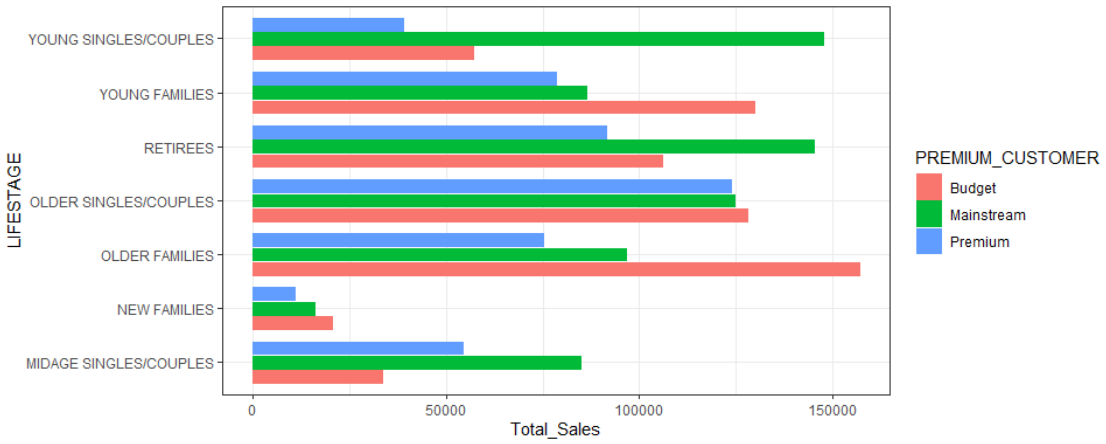
There are no nulls. So all our customers in the transaction data has been accounted for in the customer dataset.

```
fwrite(data, paste0("C:/Users/ASUS/Desktop/Quantium/","merged_data.csv"))
```

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.

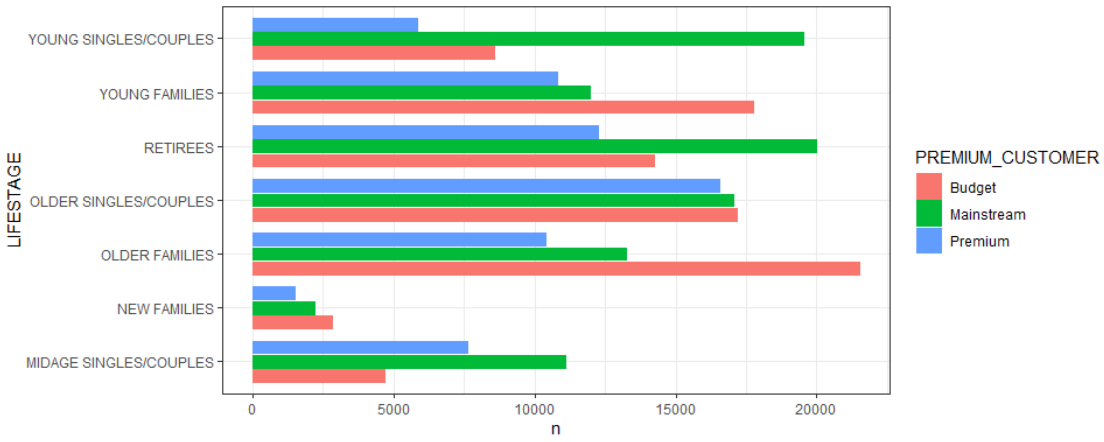
```
#### Total sales by LIFESTAGE

t = data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(Total_Sales = sum(TOT_SALES))

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

t

## # A tibble: 21 x 3
## # Groups:   LIFESTAGE [7]
##    LIFESTAGE              PREMIUM_CUSTOMER Total_Sales
##    <chr>                  <chr>                 <dbl>
##  1 MIDAGE SINGLES/COUPLES Budget                33346.
##  2 MIDAGE SINGLES/COUPLES Mainstream            84734.
##  3 MIDAGE SINGLES/COUPLES Premium               54444.
##  4 NEW FAMILIES           Budget                20607.
##  5 NEW FAMILIES           Mainstream            15980.
##  6 NEW FAMILIES           Premium               10761.
##  7 OLDER FAMILIES         Budget               156864.
##  8 OLDER FAMILIES         Mainstream            96414.
##  9 OLDER FAMILIES         Premium               75243.
## 10 OLDER SINGLES/COUPLES  Budget               127834.
## # ... with 11 more rows

### plot of total sales by lifestage
p <- ggplot(t, aes(x = LIFESTAGE, y = Total_Sales)) +
  geom_bar(
    aes(color = PREMIUM_CUSTOMER, fill = PREMIUM_CUSTOMER),
    stat = "identity", position = position_dodge(0.8),
    width = 0.7
    )+
  coord_flip()
p
```



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.

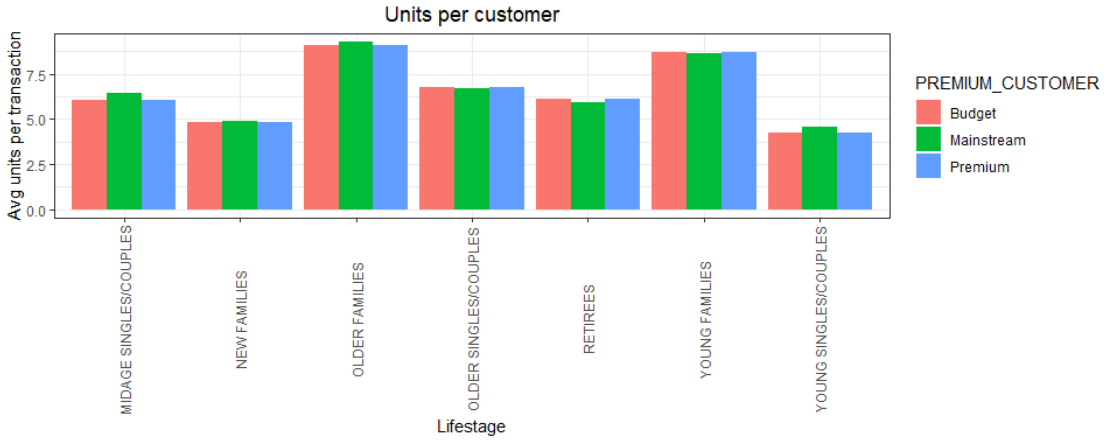
```
#### Number of customers by LIFESTAGE and PREMIUM_CUSTOMER

t = data %>%
  count(LIFESTAGE, PREMIUM_CUSTOMER)
t

##                   LIFESTAGE PREMIUM_CUSTOMER     n
##  1: MIDAGE SINGLES/COUPLES           Budget  4691
##  2: MIDAGE SINGLES/COUPLES       Mainstream 11095
##  3: MIDAGE SINGLES/COUPLES          Premium  7612
##  4:           NEW FAMILIES           Budget  2824
##  5:           NEW FAMILIES       Mainstream  2185
##  6:           NEW FAMILIES          Premium  1488
##  7:         OLDER FAMILIES           Budget 21514
##  8:         OLDER FAMILIES       Mainstream 13241
##  9:         OLDER FAMILIES          Premium 10403
## 10:  OLDER SINGLES/COUPLES           Budget 17172
## 11:  OLDER SINGLES/COUPLES       Mainstream 17061
## 12:  OLDER SINGLES/COUPLES          Premium 16560
## 13:               RETIREES           Budget 14225
## 14:               RETIREES       Mainstream 19970
## 15:               RETIREES          Premium 12236
## 16:         YOUNG FAMILIES           Budget 17763
## 17:         YOUNG FAMILIES       Mainstream 11947
## 18:         YOUNG FAMILIES          Premium 10784
## 19:  YOUNG SINGLES/COUPLES           Budget  8573
## 20:  YOUNG SINGLES/COUPLES       Mainstream 19544
## 21:  YOUNG SINGLES/COUPLES          Premium  5852
##                   LIFESTAGE PREMIUM_CUSTOMER     n

### plot of total sales by lifestage
p <- ggplot(t, aes(x = LIFESTAGE, y = n)) +
  geom_bar(
    aes(color = PREMIUM_CUSTOMER, fill = PREMIUM_CUSTOMER),
    stat = "identity", position = position_dodge(0.8),
    width = 0.7
    )+
  coord_flip()
p
```



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.

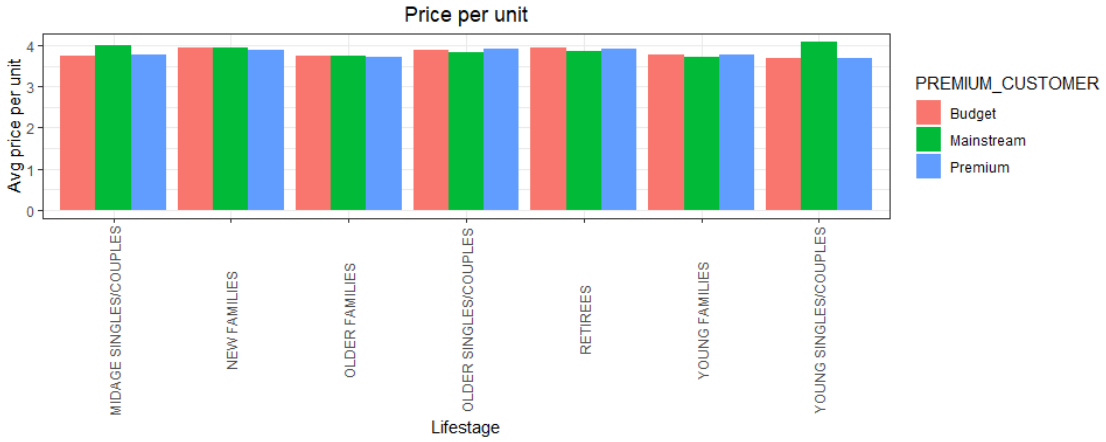
```
#### Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER
total_sales_1 <-data %>% group_by(LIFESTAGE,PREMIUM_CUSTOMER)
units <-  summarise(total_sales_1, units_count =
(sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR)))

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

summary(units)

##   LIFESTAGE         PREMIUM_CUSTOMER    units_count
##  Length:21          Length:21          Min.   :4.250
##  Class :character   Class :character   1st Qu.:4.892
##  Mode  :character   Mode  :character   Median :6.142
##                                        Mean   :6.575
##                                        3rd Qu.:8.638
##                                        Max.   :9.255

### plot of total sales by lifestage
###create plot
ggplot(data = units, aes(weight = units_count, 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))
```



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
total_sales_2 <-data %>% group_by(LIFESTAGE,PREMIUM_CUSTOMER)
pricePerUnit <-  summarise(total_sales_2, price_per_unit =
(sum(TOT_SALES)/sum(PROD_QTY)))

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

####plot
ggplot(data=pricePerUnit, aes(weight = price_per_unit,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))
```



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
pricePerUnit <- data[, price := TOT_SALES/PROD_QTY]
t.test(data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE
SINGLES/COUPLES") & PREMIUM_CUSTOMER == "Mainstream", price],data[LIFESTAGE
%in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER
!= "Mainstream", price], alternative = "greater")

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

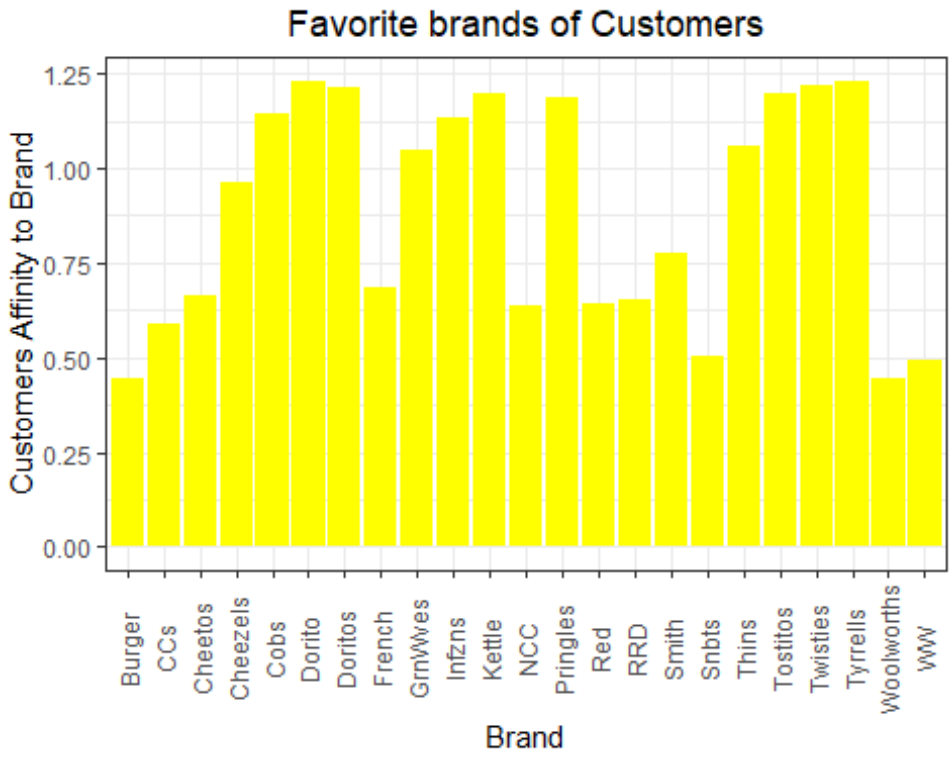
We might want to target customer segments that contribute the most to sales to retain them or further increase sales. Let's look at Mainstream - young singles/couples. For instance, let's find out if they tend to buy a particular brand of chips.

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

brand_proportions[order(-affinityToBrand)]
```

```
##          BRAND targetSegment       other affinityToBrand
##  1:     Dorito   0.015707384 0.012759861       1.2309996
##  2:   Tyrrells   0.031552795 0.025692464       1.2280953
##  3:   Twisties   0.046183575 0.037876520       1.2193194
##  4:    Doritos   0.107053140 0.088314823       1.2121764
##  5:     Kettle   0.197984817 0.165553442       1.1958967
##  6:   Tostitos   0.045410628 0.037977861       1.1957131
##  7:   Pringles   0.119420290 0.100634769       1.1866703
##  8:       Cobs   0.044637681 0.039048861       1.1431238
##  9:     Infzns   0.064679089 0.057064679       1.1334347
## 10:      Thins   0.060372671 0.056986370       1.0594230
## 11:    GrnWves   0.032712215 0.031187957       1.0488733
## 12:   Cheezels   0.017971014 0.018646902       0.9637534
## 13:      Smith   0.096369910 0.124583692       0.7735355
## 14:     French   0.003947550 0.005758060       0.6855694
## 15:    Cheetos   0.008033126 0.012066591       0.6657329
## 16:        RRD   0.032022084 0.049150801       0.6515069
## 17:        Red   0.011787440 0.018342876       0.6426168
## 18:        NCC   0.019599724 0.030853989       0.6352412
## 19:        CCs   0.011180124 0.018895650       0.5916771
## 20:      Snbts   0.006349206 0.012580210       0.5046980
## 21:         WW   0.021256039 0.043049561       0.4937574
## 22: Woolworths   0.002843340 0.006377627       0.4458304
## 23:     Burger   0.002926156 0.006596434       0.4435967
##          BRAND targetSegment       other affinityToBrand
```

```
ggplot(brand_proportions,
aes(brand_proportions$BRAND,brand_proportions$affinityToBrand)) +
geom_bar(stat = "identity",fill = "yellow") + labs(x = "Brand", y =
"Customers Affinity to Brand", title = "Favorite brands of Customers") +
theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



We can see that:

1) Mainstream young singles/couples are 23% more likely to purchase Tyrrells chips compared to the rest of the population.
2) Mainstream young singles/couples are 56% less likely to purchase Burger Rings compared to the rest of the population.

Now, find out if our target segment tends to buy larger packs of chips.

```
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
```

We can see that the preferred PACK_SIZE is 270g.

Conclusion

1) Sales have mainly been due to Budget - older families, Mainstream young singles/couples, and Mainstream - retirees shoppers.
2) 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.
3) 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.
4) 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.