Retail-Strategy-and-Analytics.R

User

2022-04-27

```
##Loading Libraries
library(data.table)
library(ggplot2)
library(ggmosaic)
library(readr)
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
```

```
library(stringr)


##Load Transaction Data
filePath <- "C:/Users/User/Downloads/QVI/"
transactionData <- fread(paste0(filePath,"QVI_transaction_data_1.csv"))

##Inspecting the transaction data
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 C
##  $ 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>
```

```r
#The transaction date is formatted incorrect.

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

##Inspecting dataset again to view changes
str(transactionData)
```

```
## Classes 'data.table' and 'data.frame':   264836 obs. of  8 variables:
##  $ DATE          : Date, format: "2018-10-17" "2019-05-14" ...
##  $ STORE_NBR     : int  1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: int  1000 1307 1343 2373 2426 4074 4149 4196 5026 7150 ...
##  $ TXN_ID        : int  1 348 383 974 1038 2982 3333 3539 4525 6900 ...
##  $ PROD_NBR      : int  5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr  "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g" "Smiths C
##  $ PROD_QTY      : int  2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num  6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
##Examining the PROD_NAME column to see that we are using the right products in this analysis
summary(transactionData$PROD_NAME)
```

```
##    Length     Class      Mode
##    264836 character character
```

```r
head(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"
```

```r
#Looks about right

##Lets ensure that they are all chips by doing some basic text analysis
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 from the set of product words.
#Removing special characters & numbers

productWords = str_replace_all(productWords$words, "[^[:alnum:]]", " ")
productWords = str_replace_all(productWords, "[[:digit:]]", "")

##Lets see all the words
product_Words <- unlist(strsplit(productWords," "))
sort(unique(product_Words), decreasing = TRUE)
```

```
##   [1] "WW"             "Woolworths"      "Whlgrn"          "Whlegrn"
##   [5] "Waves"          "Vingar"          "Vinegrg"         "Vinegr"
##   [9] "Vinegar"        "Veg"             "Tyrrells"        "Twisties"
##  [13] "Truffle"        "Tostitos"        "Tortilla"        "Tomato"
##  [17] "Tom"            "Tmato"           "Thins"           "Thinly"
##  [21] "Thai"           "Tasty"           "Tangy"           "Swt"
##  [25] "SweetChili"     "Sweet"           "Supreme"         "Sunbites"
##  [29] "Style"          "Strws"           "Sthrn"           "Steak"
##  [33] "Stacked"        "SR"              "Sr"              "Splash"
##  [37] "Spicy"          "Spcy"            "Spce"            "Sp"
##  [41] "Soy"            "Southern"        "SourCream"       "Sour"
##  [45] "Snbts"          "Snag"            "Smoked"          "Smiths"
##  [49] "Smith"          "Slt"             "Slow"            "Siracha"
##  [53] "Sensations"     "Seasonedchicken" "SeaSaltg"        "Sea"
##  [57] "Sauce"          "Salted"          "saltd"           "Salt"
##  [61] "Salsa"          "S"               "Rst"             "RRD"
##  [65] "Rock"           "Roast"           "Rings"           "Ricotta"
##  [69] "Rib"            "Red"             "Puffs"           "Pringles"
##  [73] "Prawn"          "PotatoMix"       "Potato"          "Pot"
##  [77] "Pork"           "Popd"            "Plus"            "Pesto"
##  [81] "Pepper"         "Pc"              "Paso"            "Papadums"
##  [85] "Originl"        "Original"        "Orgnl"           "OnionStacked"
##  [89] "Oniong"         "OnionDip"        "Onion"           "Onin"
##  [93] "Old"            "Of"              "NCC"             "Natural"
##  [97] "Nacho"          "N"               "Mzzrlla"         "Mystery"
## [101] "Mstrd"          "Mozzarella"      "Mild"            "Mexicana"
## [105] "Mexican"        "Medium"          "Med"             "Maple"
## [109] "Mango"          "Mac"             "Lime"            "Lightly"
## [113] "Light"          "Kettle"          "Jlpno"           "Jam"
## [117] "Jalapeno"       "Infzns"          "Infuzions"       "Htg"
## [121] "Hrb"            "Hot"             "Hony"            "Honey"
## [125] "Herbs"          "GrnWves"         "Grain"           "Gcamole"
## [129] "Garlic"         "Garden"          "G"               "g"
## [133] "Fries"          "FriedChicken"    "French"          "Frch"
## [137] "Flavour"        "Fig"             "El"              "Doritos"
## [141] "Dorito"         "Dip"             "Deli"            "D"
## [145] "CutSalt"        "Cut"             "Crnkle"          "Crnchers"
## [149] "Crn"            "Crm"             "Crisps"          "Crips"
## [153] "Crinkle"        "CreamG"          "Cream"           "Crackers"
## [157] "Corn"           "Compny"          "Coconut"         "Cobs"
## [161] "Co"             "Chutny"          "Chs"             "ChpsHny"
## [165] "ChpsFeta"       "ChpsBtroot"      "Chp"             "Chnky"
## [169] "Chlli"          "Chli"            "Chives"          "Chips"
## [173] "Chipotle"       "ChipCo"          "Chip"            "Chimuchurri"
## [177] "Chilli"         "Chili"           "Chikn"           "Chickeng"
## [181] "Chicken"        "Cheezels"        "Cheetos"         "Cheese"
## [185] "Cheddr"         "Ched"            "Chckng"          "CCs"
## [189] "Camembert"      "Burger"          "Btroot"          "Box"
## [193] "Bolognese"      "Big"             "Belly"           "BBQ"
## [197] "Basil"          "Barbeque"        "Barbecue"        "Balls"
## [201] "Bag"            "Bacon"           "And"             "Aioli"
## [205] ""
```

```
##Lets look at the most common words
wordsfreq <- data.table(unlist(product_Words))
wordsfreq
```

```
##           V1
##   1: Natural
##   2:    Chip
##   3:
##   4:
##   5:
##  ---
## 863: Doritos
## 864:   Salsa
## 865:    Mild
## 866:
## 867:       g
```

```
##There are salsa products in the dataset but we are only interested in the chips category.
#so let's remove these.
transactionData[!grepl("salsa", tolower(transactionData$PROD_NAME)),]
```

```
##               DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##      1: 2018-10-17         1           1000      1        5
##      2: 2019-05-14         1           1307    348       66
##      3: 2019-05-20         1           1343    383       61
##      4: 2018-08-17         2           2373    974       69
##      5: 2018-08-18         2           2426   1038      108
##     ---
## 246738: 2019-03-09       272         272319 270088       89
## 246739: 2018-08-13       272         272358 270154       74
## 246740: 2018-11-06       272         272379 270187       51
## 246741: 2018-12-27       272         272379 270188       42
## 246742: 2018-09-22       272         272380 270189       74
##                                      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
##     ---
## 246738:  Kettle Sweet Chilli And Sour Cream 175g        2      10.8
## 246739:            Tostitos Splash Of  Lime 175g        1       4.4
## 246740:                 Doritos Mexicana    170g        2       8.8
## 246741:  Doritos Corn Chip Mexican Jalapeno 150g        2       7.8
## 246742:            Tostitos Splash Of  Lime 175g        2       8.8
```

```
##Now let's see our transaction data again
str(transactionData)
```

```
## Classes 'data.table' and 'data.frame':   264836 obs. of  8 variables:
##  $ DATE          : Date, format: "2018-10-17" "2019-05-14" ...
##  $ STORE_NBR     : int  1 1 1 2 2 4 4 4 5 7 ...
```

```
##  $ LYLTY_CARD_NBR: int  1000 1307 1343 2373 2426 4074 4149 4196 5026 7150 ...
##  $ TXN_ID        : int  1 348 383 974 1038 2982 3333 3539 4525 6900 ...
##  $ PROD_NBR      : int  5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr  "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g" "Smiths C
##  $ PROD_QTY      : int  2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num  6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
##Checking the summary statistiscs to see if there are any obvious outliers or missing values (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.:  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
```

```
##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.

##Lets find the outlier
transactionData %>% filter(transactionData$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
##both of these transactions were by the same customer with loyalty card number 226000.

##Lets see if this customer made other transactions
transactionData %>% filter(transactionData$LYLTY_CARD_NBR == 226000, transactionData$TXN_ID != 226201 |
```

```
##          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
```

```
##They didn't.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.

##Let's remove this loyalty card number from further analysis.
transactionData[grepl(226000, transactionData$LYLTY_CARD_NBR),]
```

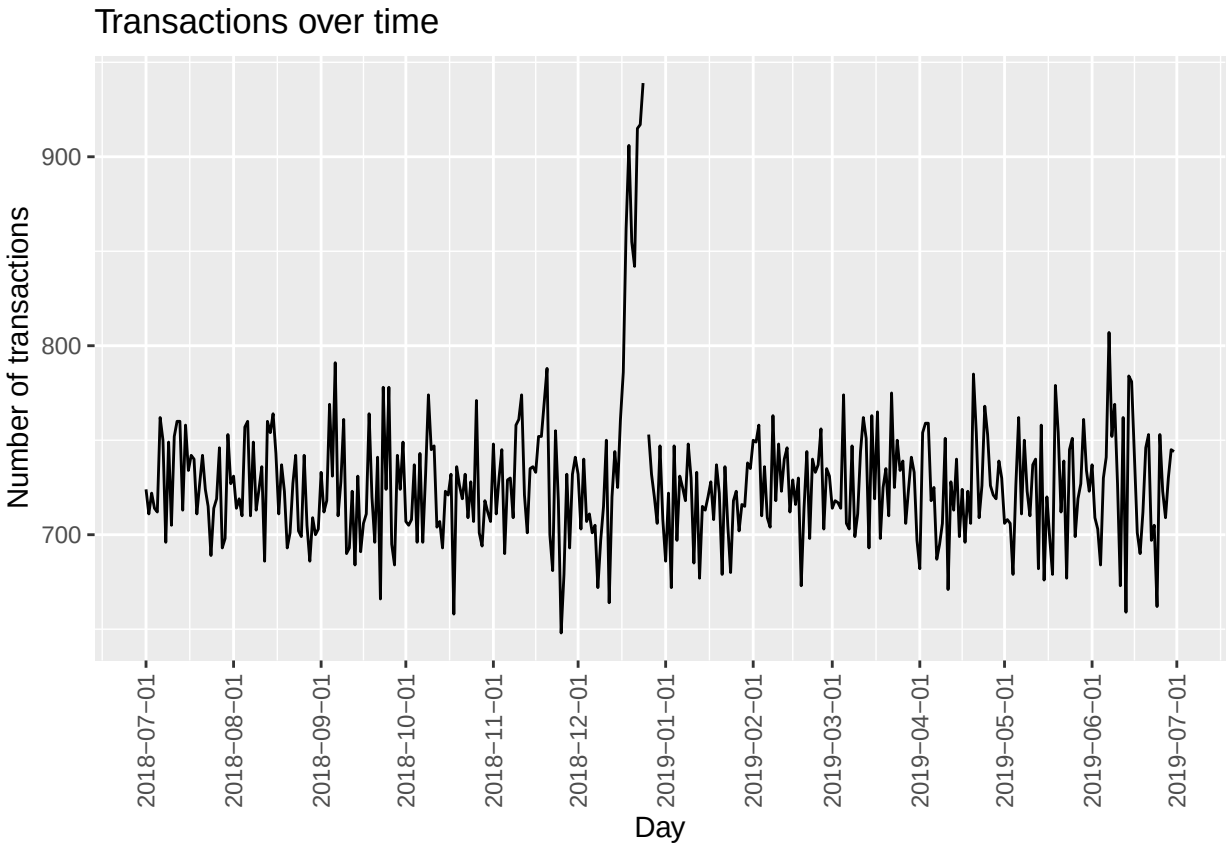
```
##          DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 1: 2018-08-19       226         226000 226201        4
## 2: 2019-05-20       226         226000 226210        4
##                          PROD_NAME PROD_QTY TOT_SALES
## 1: Dorito Corn Chp     Supreme 380g      200       650
## 2: Dorito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

```
transactionData <- transactionData %>% filter(transactionData$LYLTY_CARD_NBR!= 226000)

##Examing the dataset again to view changes
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.:  70021   1st Qu.:  67601
##  Median :2018-12-30   Median :130.0   Median : 130357   Median : 135137
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135549   Mean   : 135158
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202700
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:264834      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   :1.906   Mean   : 7.299
##  3rd Qu.: 85.00                      3rd Qu.:2.000   3rd Qu.: 9.200
##  Max.   :114.00                      Max.   :5.000   Max.   :29.500
```

```
##Counting the number of transactions by date
transactions_by_day = transactionData %>% group_by(DATE) %>% summarise(N = n())

##There's only 364 rows, meaning only 364 dates which indicates a missing date.
##Let's find the missing date
date_range <- seq(min(transactions_by_day$DATE), max(transactions_by_day$DATE), by = 1)
missingDay <- date_range[!date_range %in% transactions_by_day$DATE]
##The missing day is 2018-12-52, Christmas Day
##This implies that there are zero sales on Christmas day itself.
##This is likely due to shops being closed on Christmas day.

## Creatimg a sequence of dates and join this the count of transactions by date
allDates <- data.table(seq(as.Date("2018/07/01"), as.Date("2019/06/30"), by = "day"))
setnames(allDates, "DATE")
transactions_by_day <- merge(allDates, transactionData[, .N, by = DATE], all.x = TRUE)

##Plotting Transactions over time
transactions_by_day %>% ggplot(aes(x = DATE, y = N)) +
```

```
  geom_line() +
  labs(x = "Day", y = "Number of transactions", title = "Transactions over time") +
  scale_x_date(breaks = "1 month") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

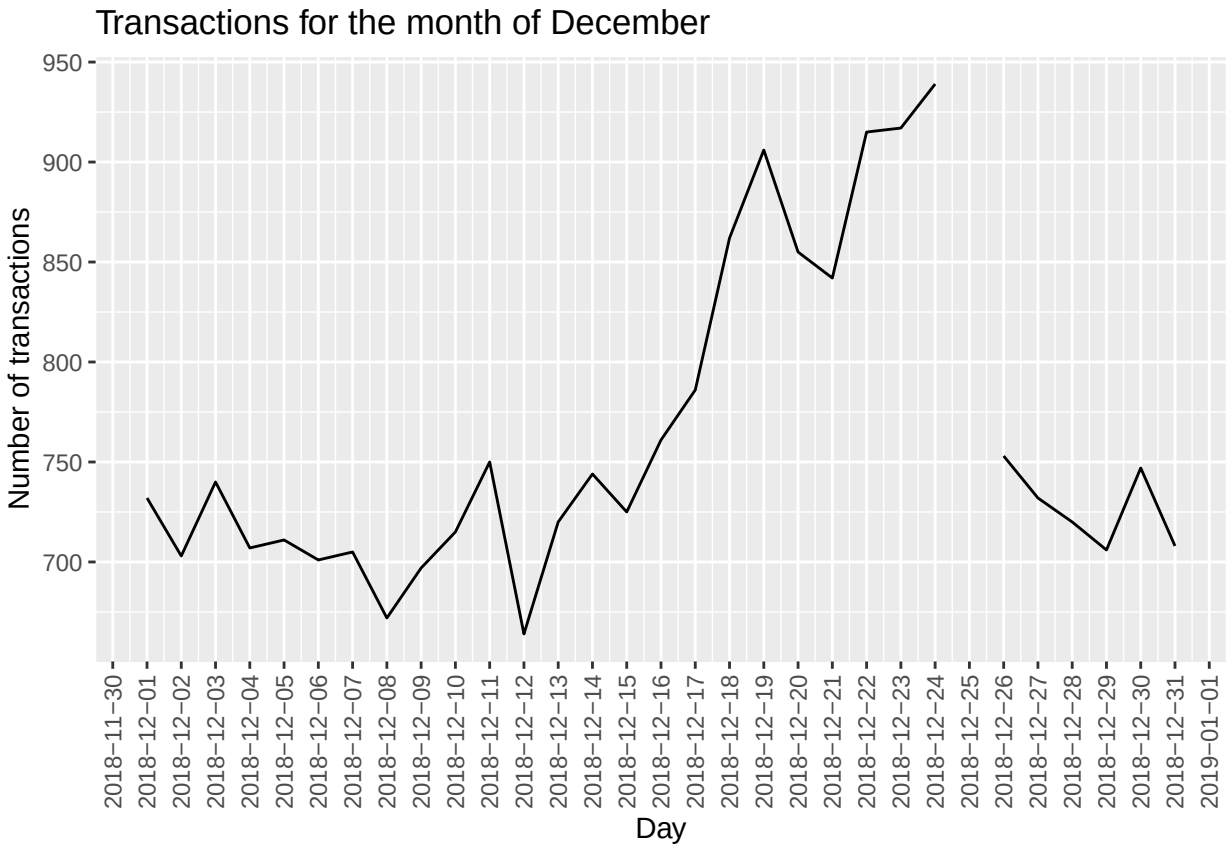



```
##We can see that there is an increase in purchases in December and a break in late
##December. Let's zoom in on this.
transactions_by_day %>% filter(month(DATE) == 12)
```

```
##           DATE   N
##  1: 2018-12-01 732
##  2: 2018-12-02 703
##  3: 2018-12-03 740
##  4: 2018-12-04 707
##  5: 2018-12-05 711
##  6: 2018-12-06 701
##  7: 2018-12-07 705
##  8: 2018-12-08 672
##  9: 2018-12-09 697
## 10: 2018-12-10 715
## 11: 2018-12-11 750
## 12: 2018-12-12 664
## 13: 2018-12-13 720
## 14: 2018-12-14 744
## 15: 2018-12-15 725
```

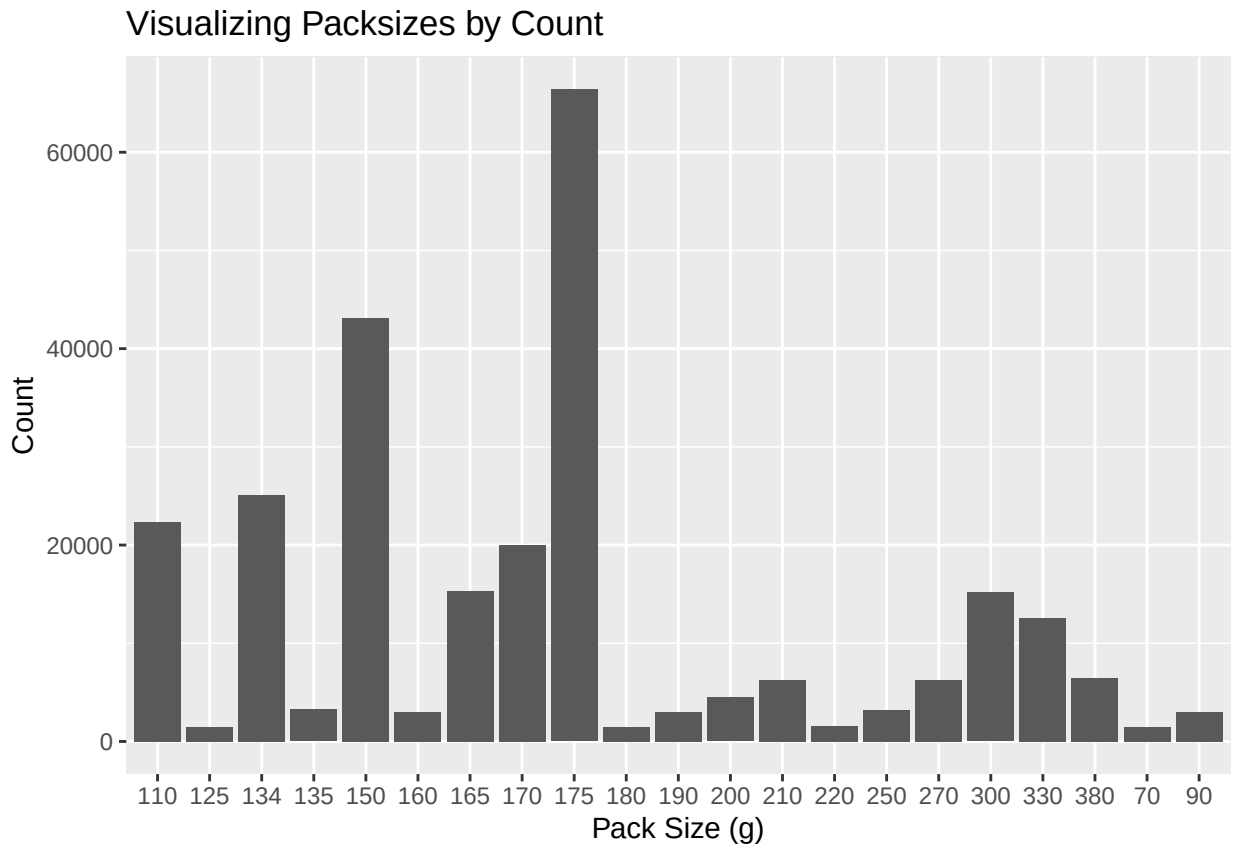
```
## 16: 2018-12-16 761
## 17: 2018-12-17 786
## 18: 2018-12-18 862
## 19: 2018-12-19 906
## 20: 2018-12-20 855
## 21: 2018-12-21 842
## 22: 2018-12-22 915
## 23: 2018-12-23 917
## 24: 2018-12-24 939
## 25: 2018-12-25  NA
## 26: 2018-12-26 753
## 27: 2018-12-27 732
## 28: 2018-12-28 720
## 29: 2018-12-29 706
## 30: 2018-12-30 747
## 31: 2018-12-31 708
##           DATE   N
```

```
transactions_by_day %>% filter(month(DATE) == 12) %>% ggplot(aes(x = DATE, y = N)) +
  geom_line() +
  labs(x = "Day", y = "Number of transactions", title = "Transactions for the month of December") +
  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

##Comverting from data frame to data table
setDT(transactionData)

##Creating an augumented colomn "PACK_SIZE" from PROD_NAME by taking the digits.
transactionData[, PACK_SIZE := parse_number(PROD_NAME)]
Packsizes <- transactionData[, .N, PACK_SIZE][order(PACK_SIZE)]
##The largest size is 380g and the smallest size is 70g - seems sensible!

##plotting a histogram of PACK_SIZE since we know that it is a categorical variable
##and not a continuous variable even though it is numeric.
Packsizes %>% ggplot(aes(x = as.character(PACK_SIZE), y = Packsizes$N)) +
  geom_histogram(stat = "identity", bins = 5) +
  labs(x = "Pack Size (g)", y = "Count", title = "Visualizing Packsizes by Count")
```

```
## Warning: Ignoring unknown parameters: binwidth, bins, pad
```




```
##Creating an augumented colomn "Brand" from PROD_NAME by taking the the first words.
transactionData$Brand <- word(transactionData$PROD_NAME, 1)
##Let's see the brand names
head(transactionData$Brand)
```

```
## [1] "Natural" "CCs"     "Smiths"  "Smiths"  "Kettle"  "Old"
```

```
##Let's check that the brand names are sensible
sort(unique(transactionData$Brand), decreasing = FALSE)
```

```
##  [1] "Burger"     "CCs"        "Cheetos"    "Cheezels"   "Cobs"
##  [6] "Dorito"     "Doritos"    "French"     "Grain"      "GrnWves"
## [11] "Infuzions"  "Infzns"     "Kettle"     "Natural"    "NCC"
## [16] "Old"        "Pringles"   "Red"        "RRD"        "Smith"
## [21] "Smiths"     "Snbts"      "Sunbites"   "Thins"      "Tostitos"
## [26] "Twisties"   "Tyrrells"   "Woolworths" "WW"
```

```
##Some of the brand names look like they are of the same brands. We will combine these
transactionData[Brand == "Dorito", Brand := "Doritos"]
transactionData[Brand == "Natural", Brand := "NCC"]
transactionData[Brand == "Red", Brand := "RRD"]
transactionData[Brand == "Grain", Brand := "GrainWaves"]
transactionData[Brand == "GrnWves", Brand := "GrainWaves"]
transactionData[Brand == "Infzns", Brand := "Infuzions"]
transactionData[Brand == "Smith", Brand := "Smiths"]
transactionData[Brand == "Snbts", Brand := "Sunbites"]
transactionData[Brand == "WW", Brand := "Woolworths"]

##Re-examing the brand names
sort(unique(transactionData$Brand), decreasing = FALSE)
```

```
##  [1] "Burger"     "CCs"        "Cheetos"    "Cheezels"   "Cobs"
##  [6] "Doritos"    "French"     "GrainWaves" "Infuzions"  "Kettle"
## [11] "NCC"        "Old"        "Pringles"   "RRD"        "Smiths"
## [16] "Sunbites"   "Thins"      "Tostitos"   "Twisties"   "Tyrrells"
## [21] "Woolworths"
```

```
##That looks about right! Let's re-examine our dataset
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.:  70021   1st Qu.:  67601
##  Median :2018-12-30   Median :130.0   Median : 130357   Median : 135137
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135549   Mean   : 135158
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202700
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:264834      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   :1.906   Mean   : 7.299
##  3rd Qu.: 85.00                      3rd Qu.:2.000   3rd Qu.: 9.200
##  Max.   :114.00                      Max.   :5.000   Max.   :29.500
##    PACK_SIZE        Brand
##  Min.   : 70.0   Length:264834
##  1st Qu.:150.0   Class :character
##  Median :170.0   Mode  :character
##  Mean   :182.4
```

```
##  3rd Qu.:175.0
##  Max.   :380.0
```

```
##Now that we are happy with the transaction dataset
##let's have a look at the customer behavior dataset.
#Loading data...
QVI_purchase_behaviour_1_ <-  fread(paste0(filePath,"QVI_purchase_behaviour_1.csv"))
summary(QVI_purchase_behaviour_1_)
```

```
##  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
```

```
##JOining both datasets (LEFT JOIN)
data <- merge(transactionData, QVI_purchase_behaviour_1_, 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.

##Examining the new dataset to ensure that all customers are accounted for i.e., there are no nulls.
summary(data)
```

```
##  LYLTY_CARD_NBR         DATE              STORE_NBR         TXN_ID
##  Min.   :   1000   Min.   :2018-07-01   Min.   :  1.0   Min.   :      1
##  1st Qu.:  70021   1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  67601
##  Median : 130357   Median :2018-12-30   Median :130.0   Median : 135137
##  Mean   : 135549   Mean   :2018-12-30   Mean   :135.1   Mean   : 135158
##  3rd Qu.: 203094   3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 202700
##  Max.   :2373711   Max.   :2019-06-30   Max.   :272.0   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:264834      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   :1.906   Mean   : 7.299
##  3rd Qu.: 85.00                      3rd Qu.:2.000   3rd Qu.: 9.200
##  Max.   :114.00                      Max.   :5.000   Max.   :29.500
##    PACK_SIZE        Brand             LIFESTAGE         PREMIUM_CUSTOMER
##  Min.   : 70.0   Length:264834      Length:264834      Length:264834
##  1st Qu.:150.0   Class :character   Class :character   Class :character
##  Median :170.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :182.4
##  3rd Qu.:175.0
##  Max.   :380.0
```

```
##Checking for nulls.
sum(is.na(data$LIFESTAGE))
```

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

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

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

```
##Now that the data is clean and ready for analysis. Let's save it for future reference
#fwrite(data, paste0("C:/Users/User/Downloads/","Quantum_data_cleaned.xlsx"))

##Data exploration is now complete!

###Data Analysis on customer segments

##Examining the Lifestage and Premium Customer segments
unique(data$LIFESTAGE)
```

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

```
unique(data$PREMIUM_CUSTOMER)
```

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

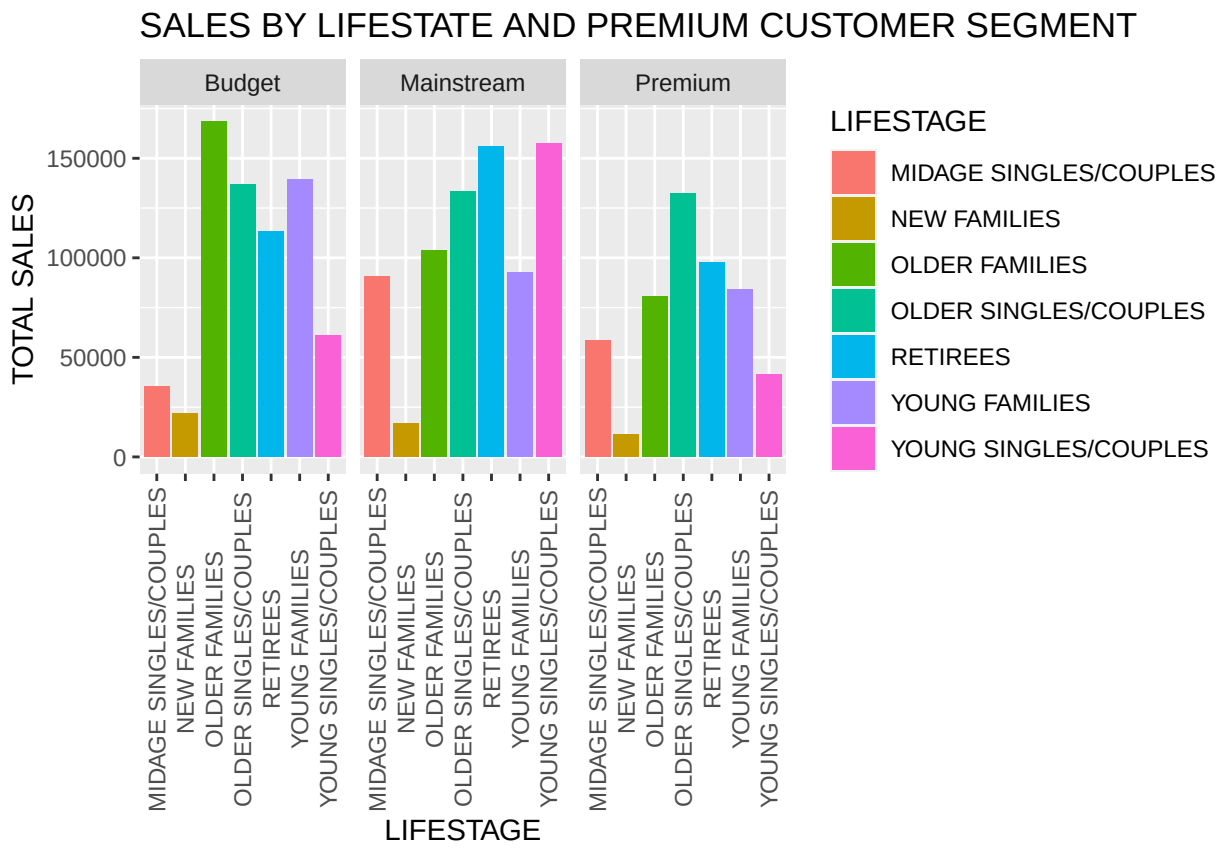
```
####We can answer some key questions.
###1. Who spends the most on chips (total sales)?
###describing customers by lifestage and how premium their general purchasing behaviour is.

##Summarizing total sales by each lifestage and premium customer segment
data %>% group_by(LIFESTAGE) %>% summarize(sum = sum(TOT_SALES))
```

```
## # A tibble: 7 x 2
##   LIFESTAGE                  sum
##   <chr>                    <dbl>
## 1 MIDAGE SINGLES/COUPLES 184751.
## 2 NEW FAMILIES            50433.
## 3 OLDER FAMILIES         352467.
## 4 OLDER SINGLES/COUPLES  402427.
## 5 RETIREES               366471.
## 6 YOUNG FAMILIES         316160.
## 7 YOUNG SINGLES/COUPLES  260405.
```

```
data %>% group_by(PREMIUM_CUSTOMER) %>% summarize(sum = sum(TOT_SALES))
```

```
## # A tibble: 3 x 2
##   PREMIUM_CUSTOMER     sum
##   <chr>              <dbl>
## 1 Budget           676212.
## 2 Mainstream       750744.
## 3 Premium          506159.
```

```
##Ploting sales by each lifestage and premium customer segments
data %>% ggplot(aes(x = LIFESTAGE, y = TOT_SALES, fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~data$PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "LIFESTAGE", y = "TOTAL SALES", title = "SALES BY LIFESTATE AND PREMIUM CUSTOMER SEGMENT")
```




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


###2. How many customers are in each segment?

##Calculate number of Unique customers
n_distinct(data$LYLTY_CARD_NBR)
```

```
## [1] 72636
```

```
##Summarizing number of customers by lifestage and premium customer segments
data %>% group_by(LIFESTAGE) %>% summarise(no_of_customers = n_distinct(LYLTY_CARD_NBR))
```

```
## # A tibble: 7 x 2
##   LIFESTAGE              no_of_customers
##   <chr>                            <int>
## 1 MIDAGE SINGLES/COUPLES            7275
```

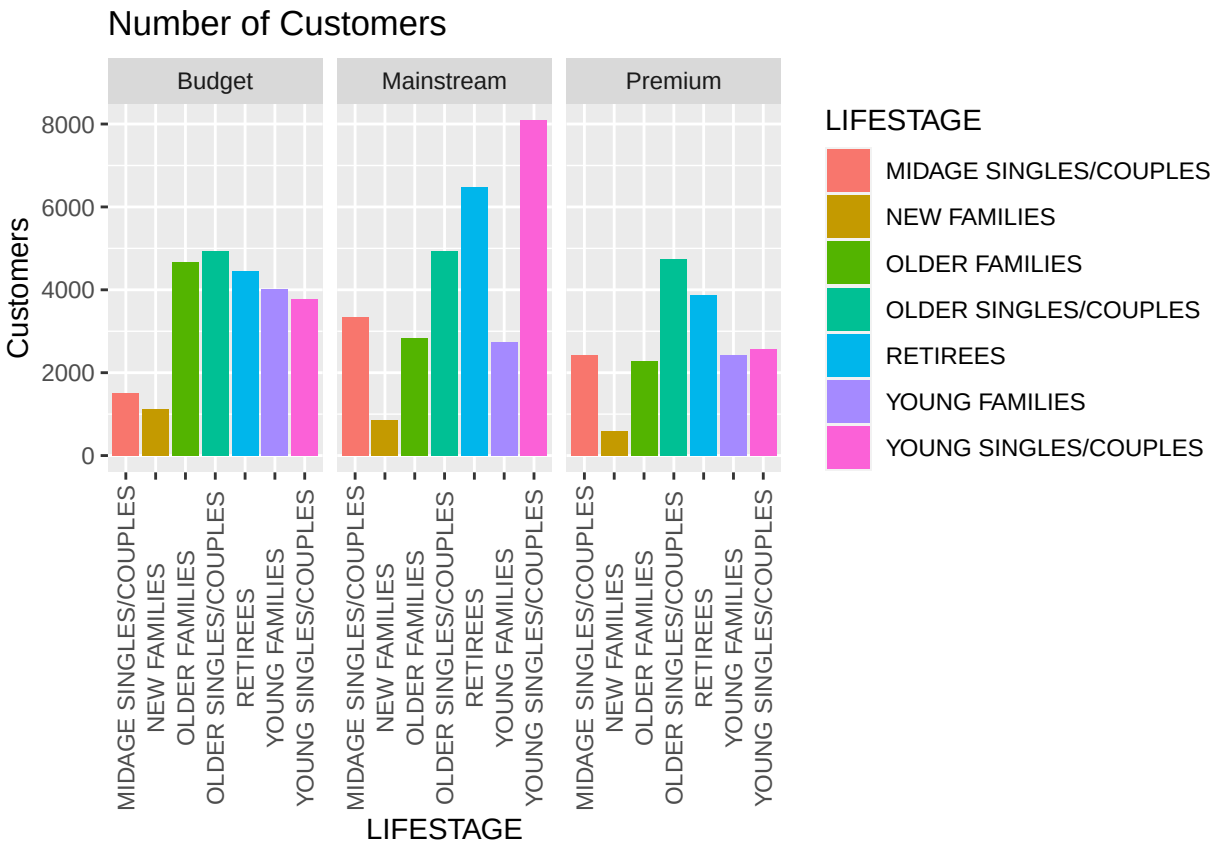
```
## 2 NEW FAMILIES                       2549
## 3 OLDER FAMILIES                     9779
## 4 OLDER SINGLES/COUPLES             14609
## 5 RETIREES                          14805
## 6 YOUNG FAMILIES                     9178
## 7 YOUNG SINGLES/COUPLES             14441
```

```
data %>% group_by(PREMIUM_CUSTOMER) %>% summarise(no_of_customers = n_distinct(LYLTY_CARD_NBR))
```

```
## # A tibble: 3 x 2
##   PREMIUM_CUSTOMER no_of_customers
##   <chr>                      <int>
## 1 Budget                     24470
## 2 Mainstream                 29245
## 3 Premium                    18921
```

```
data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>% summarise(ncus = n_distinct(LYLTY_CARD_NBR)) %>%
  ggplot(aes(x = LIFESTAGE, y = ncus, fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  facet_grid(~PREMIUM_CUSTOMER) +
  labs(x = "LIFESTAGE", y = "Customers", title = "Number of Customers")
```

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



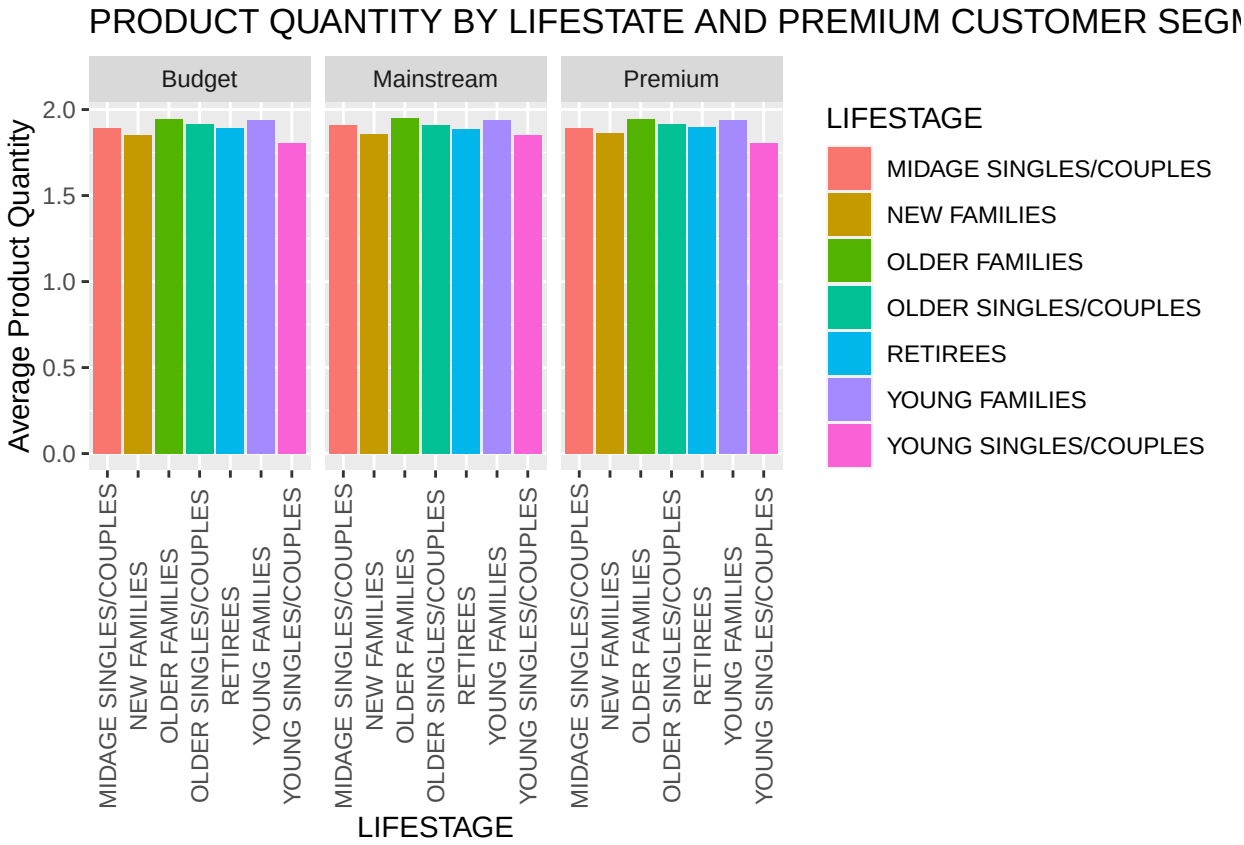
```
##There are more Mainstream - young singles/couples and Mainstream - retirees who buy chips.
##This contributes to there being more sales to these customer segments
##but this is not a major driver for the Budget - Older families segment.

##Higher sales may also be driven by more units of chips being bought per customer
##Calculating the average product quantity for each segment
data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>% summarise(avg_product_quantity = mean(PROD_QTY))
```

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

```
## # A tibble: 21 x 3
## # Groups:   LIFESTAGE [7]
##    LIFESTAGE              PREMIUM_CUSTOMER avg_product_quantity
##    <chr>                  <chr>                           <dbl>
##  1 MIDAGE SINGLES/COUPLES Budget                           1.89
##  2 MIDAGE SINGLES/COUPLES Mainstream                       1.91
##  3 MIDAGE SINGLES/COUPLES Premium                          1.89
##  4 NEW FAMILIES           Budget                           1.85
##  5 NEW FAMILIES           Mainstream                       1.86
##  6 NEW FAMILIES           Premium                          1.86
##  7 OLDER FAMILIES         Budget                           1.95
##  8 OLDER FAMILIES         Mainstream                       1.95
##  9 OLDER FAMILIES         Premium                          1.95
## 10 OLDER SINGLES/COUPLES  Budget                           1.91
## # ... with 11 more rows
```

```
data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>% summarise(avg_product_quantity = mean(PROD_QTY)) %>%
  ggplot(aes(x = LIFESTAGE, y = avg_product_quantity , fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "LIFESTAGE", y = "Average Product Quantity",
       title = "PRODUCT QUANTITY BY LIFESTATE AND PREMIUM CUSTOMER SEGMENT")
```

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

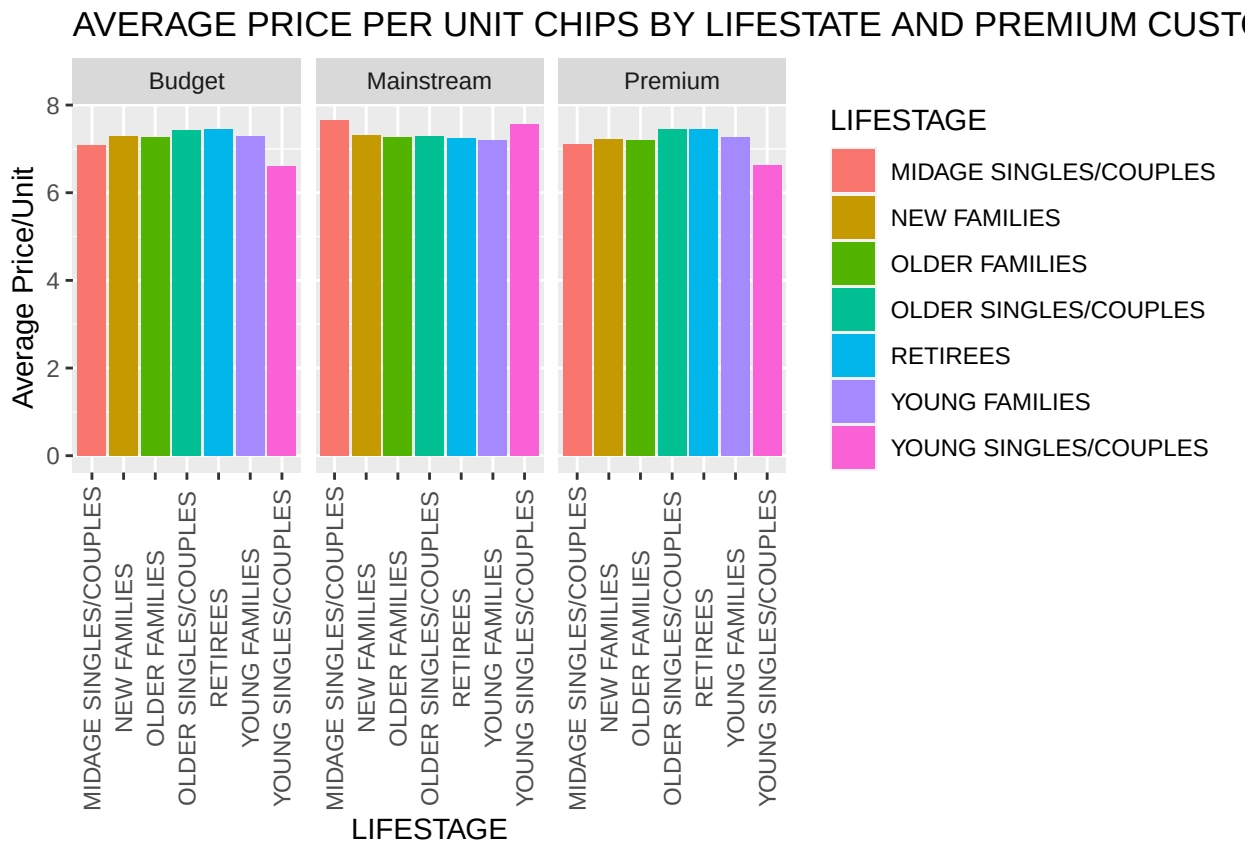



```
#Older families and young families in general buy more chips per customer

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

data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>% summarise(avg_unit_price = mean(TOT_SALES))
```

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

## # A tibble: 21 x 3
## # Groups:   LIFESTAGE [7]
##    LIFESTAGE              PREMIUM_CUSTOMER avg_unit_price
##    <chr>                  <chr>                     <dbl>
##  1 MIDAGE SINGLES/COUPLES Budget                     7.07
##  2 MIDAGE SINGLES/COUPLES Mainstream                 7.65
##  3 MIDAGE SINGLES/COUPLES Premium                    7.11
##  4 NEW FAMILIES           Budget                     7.30
##  5 NEW FAMILIES           Mainstream                 7.32
##  6 NEW FAMILIES           Premium                    7.23
##  7 OLDER FAMILIES         Budget                     7.27
##  8 OLDER FAMILIES         Mainstream                 7.26
##  9 OLDER FAMILIES         Premium                    7.21
## 10 OLDER SINGLES/COUPLES  Budget                     7.43
## # ... with 11 more rows
```

```
data %>% group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>% summarise(avg_unit_price = mean(TOT_SALES)) %>%
  ggplot(aes(x = LIFESTAGE, y = avg_unit_price , fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "LIFESTAGE", y = "Average Price/Unit",
       title = "AVERAGE PRICE PER UNIT CHIPS BY LIFESTATE AND PREMIUM CUSTOMER SEGMENT")
```

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




```
##Mainstream midage and young singles/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.
### Performing an independent t-test between mainstream vs premium and budget midage and young singles/
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")
```

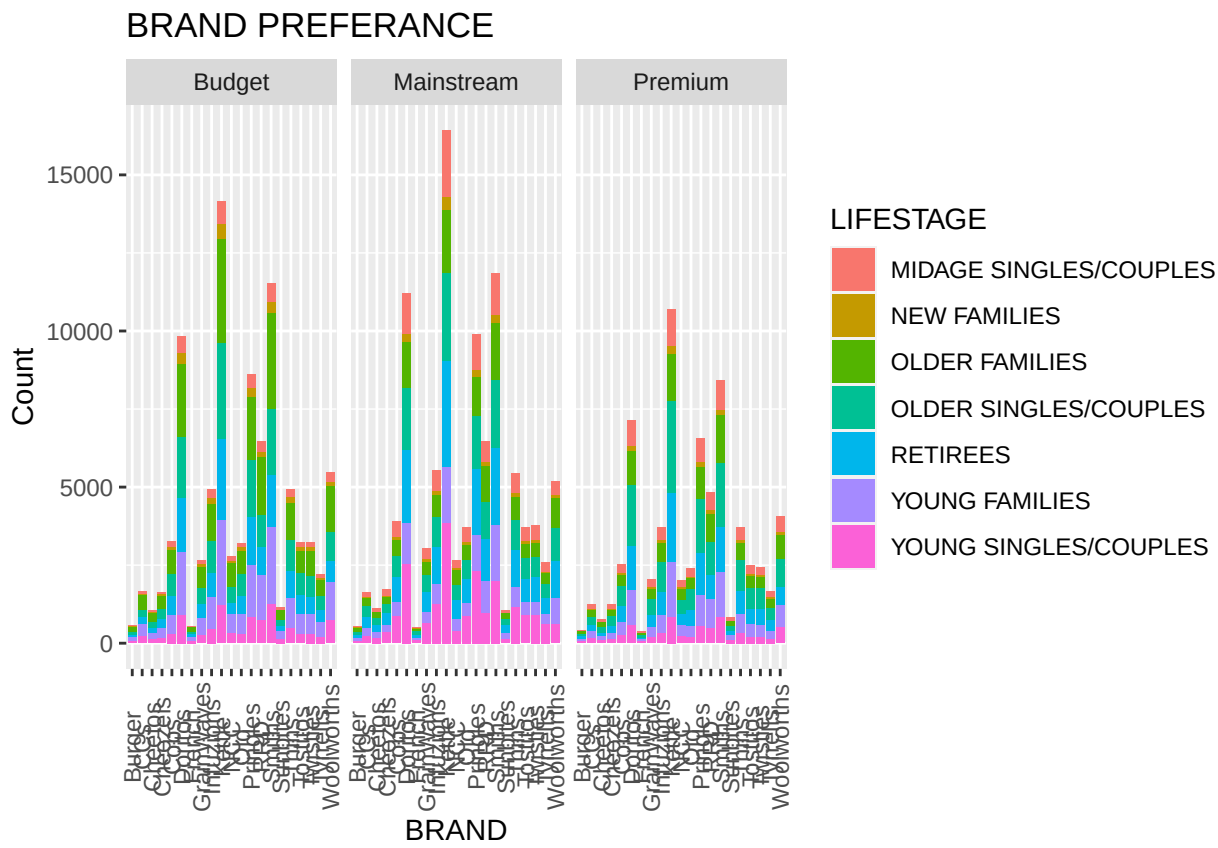
```
##
##  Welch Two Sample t-test
##
## data:  data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER ==
## t = 40.61, df = 58792, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3429435       Inf
## sample estimates:
## mean of x mean of y
##  4.045586  3.688165
```

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

##We can dive deeper into customer segments for insights.
##We can target customer segments that contribute the most to sales to retain them or further increase
##For example, let's look at Mainstream - young singles/couples and find out if they tend to buy a part

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 := ta
brand_proportions[order(-affinityToBrand)]
```

```
##          Brand targetSegment       other affinityToBrand
##  1:   Tyrrells   0.029586871 0.023933043       1.2362352
##  2:   Twisties   0.043306068 0.035282734       1.2274011
##  3:     Kettle   0.185649203 0.154216335       1.2038232
##  4:   Tostitos   0.042581280 0.035377136       1.2036384
##  5:        Old   0.041597639 0.034752796       1.1969581
##  6:   Pringles   0.111979706 0.093743295       1.1945356
##  7:    Doritos   0.122877407 0.105277499       1.1671764
##  8:       Cobs   0.041856492 0.036374793       1.1507005
##  9:  Infuzions   0.060649203 0.053156887       1.1409472
## 10:      Thins   0.056611100 0.053083941       1.0664449
## 11: GrainWaves   0.030674053 0.029052204       1.0558253
## 12:   Cheezels   0.016851315 0.017369961       0.9701412
## 13:     Smiths   0.093419963 0.121714168       0.7675356
## 14:     French   0.003701595 0.005363748       0.6901134
## 15:    Cheetos   0.007532615 0.011240270       0.6701454
## 16:        RRD   0.045376890 0.068426405       0.6631488
## 17:        NCC   0.018378546 0.028741107       0.6394516
## 18:        CCs   0.010483537 0.017601675       0.5955988
## 19:   Sunbites   0.005953614 0.011718716       0.5080431
## 20: Woolworths   0.028189066 0.057428576       0.4908543
## 21:     Burger   0.002743839 0.006144710       0.4465369
##          Brand targetSegment       other affinityToBrand
```

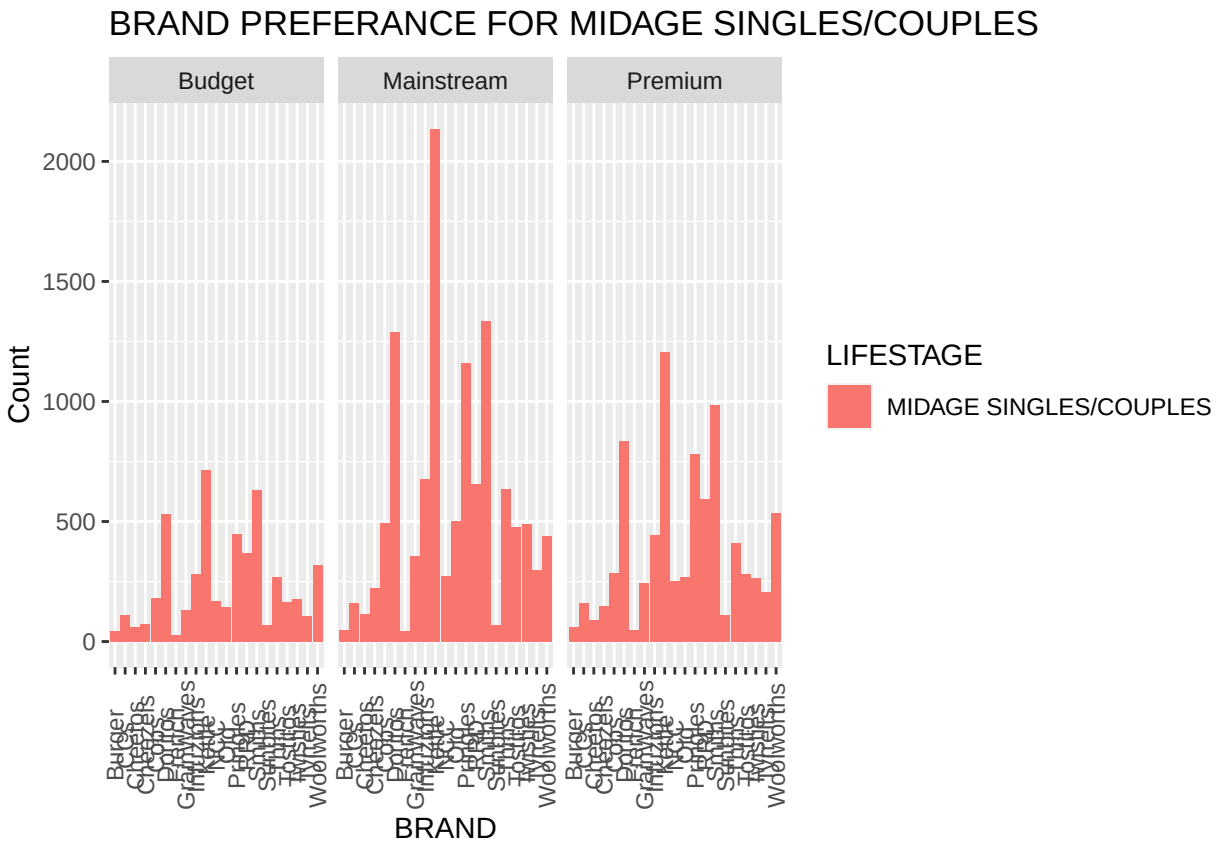
```
##Mainstream young singles/couples are 23% more likely to purchase Tyrrells chips compared to the rest
##Mainstream young singles/couples are 56% less likely to purchase Burger Rings compared to the rest of


##We can also check if there are brands that all customer segments prefer to others
data %>% group_by(Brand, LIFESTAGE, PREMIUM_CUSTOMER) %>% summarise(num = n()) %>%
  ggplot(aes(x = Brand, y = num, fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "BRAND", y = "Count", title = "BRAND PREFERANCE")
```

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



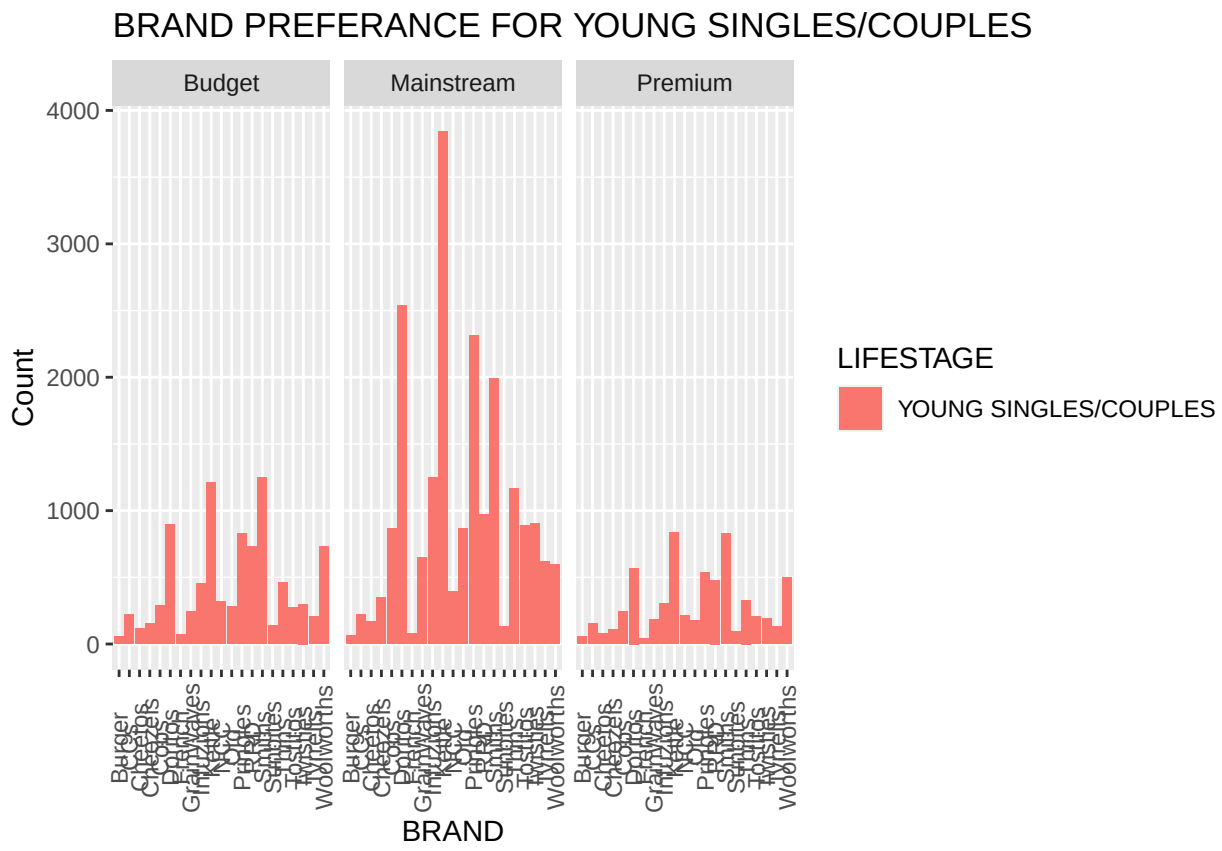

```
##There is a general prefence of the brand - Kettle accross premium customer segments.
##We zoom in and see what brands are preffered by the lifestage segments that
##are more willing to pay more per packet of chips

##Let's see what brands are preferred by midage singles and couples
data %>% group_by(Brand, LIFESTAGE, PREMIUM_CUSTOMER) %>% filter(LIFESTAGE == "MIDAGE SINGLES/COUPLES")
  summarise(num = n()) %>% ggplot(aes(x = Brand, y = num, fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "BRAND", y = "Count", title = "BRAND PREFERANCE FOR MIDAGE SINGLES/COUPLES")
```

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

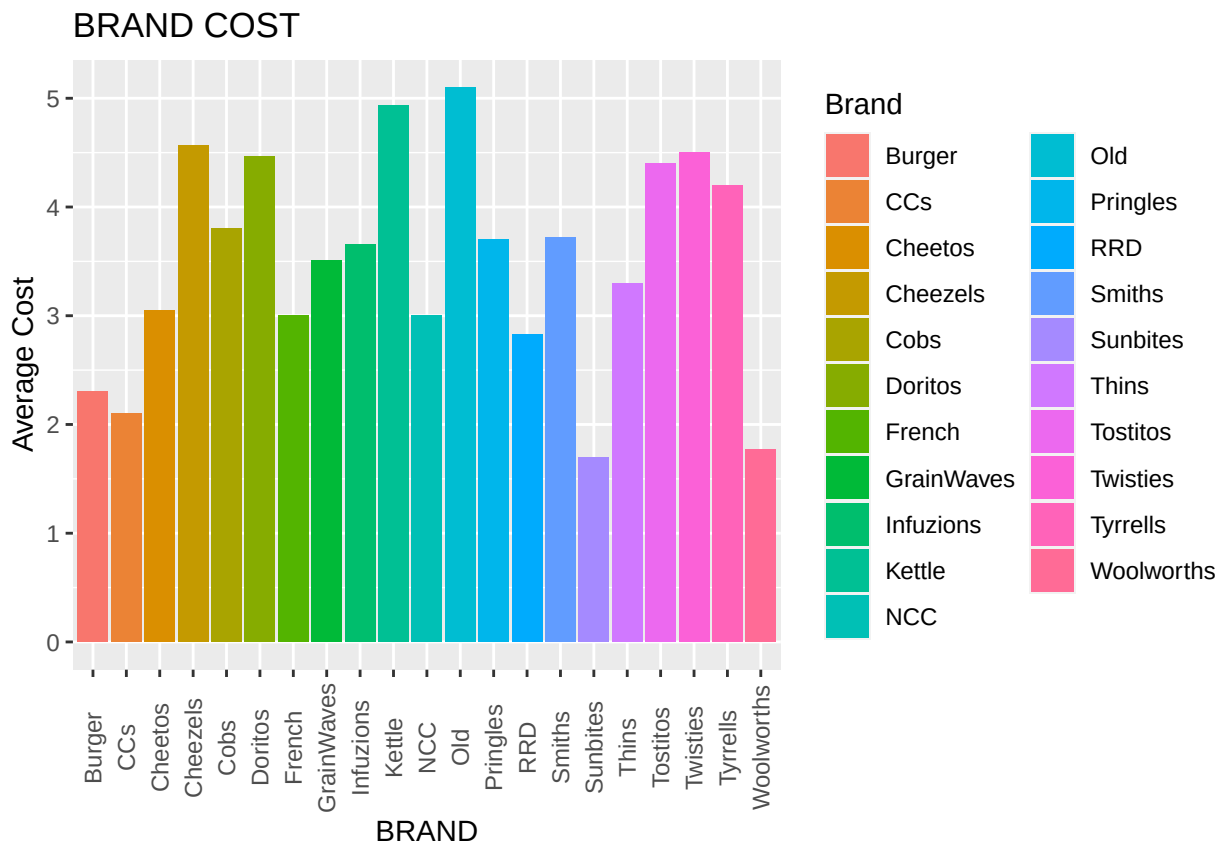



```
##Midage singles generally prefer Kettle chips, especially mainstream customers.
##Mainsteam customers in this segment also like Doritos, Smiths, and Pringles respectively.

##Let's see what brands are preferred by young singles and couples
data %>% group_by(Brand, LIFESTAGE, PREMIUM_CUSTOMER) %>% filter(LIFESTAGE == "YOUNG SINGLES/COUPLES") %
  summarise(num = n()) %>% ggplot(aes(x = Brand, y = num, fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "BRAND", y = "Count", title = "BRAND PREFERANCE FOR YOUNG SINGLES/COUPLES")
```

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

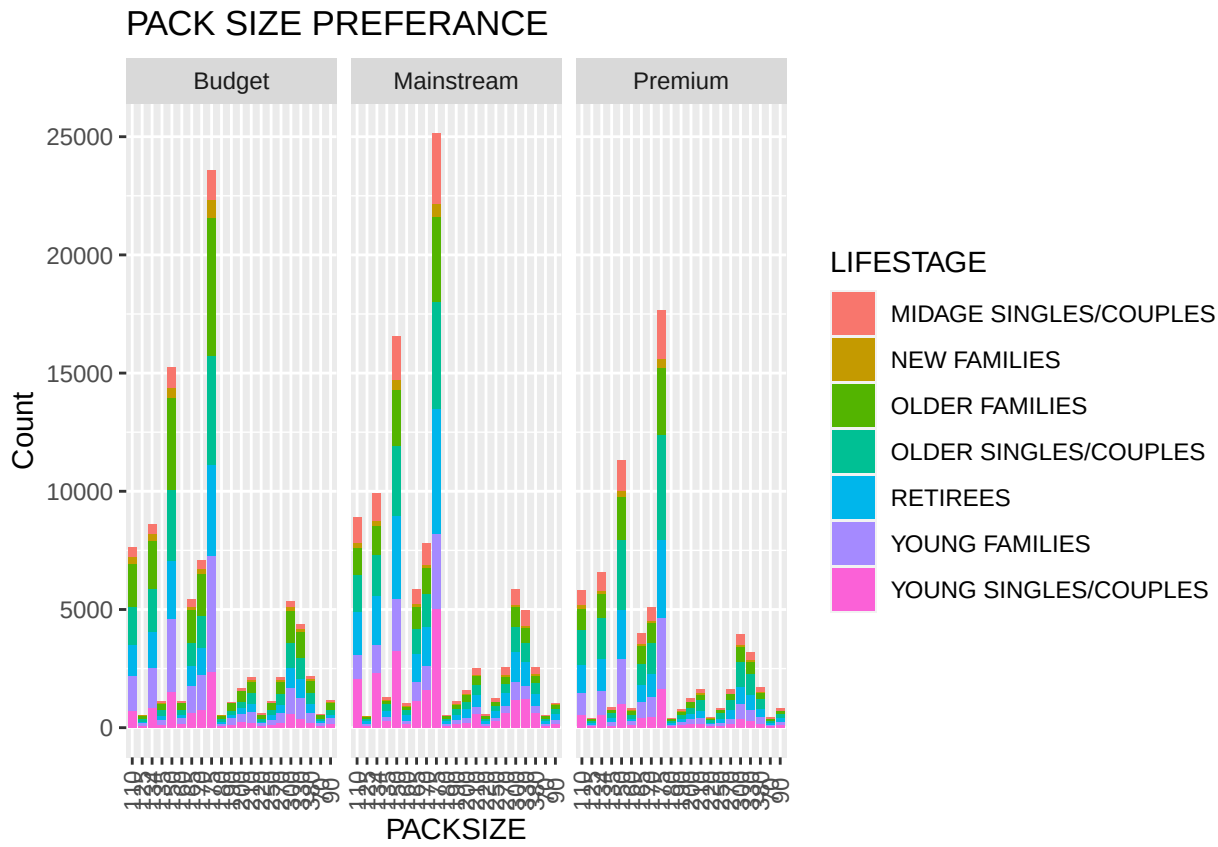



```
##Young singles generally prefer Kettle chips, especially mainstream customers.
##Mainsteam customers in this segment also like Doritos, Pringles and Smiths respectively.

##Let's see if the preference for Kettle chips is a result of low or high price
data %>% group_by(Brand) %>% summarise(cost = mean(TOT_SALES/PROD_QTY)) %>%
  ggplot(aes(x = Brand, y = cost, fill = Brand)) +
  geom_bar(stat ="identity") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "BRAND", y = "Average Cost", title = "BRAND COST")
```



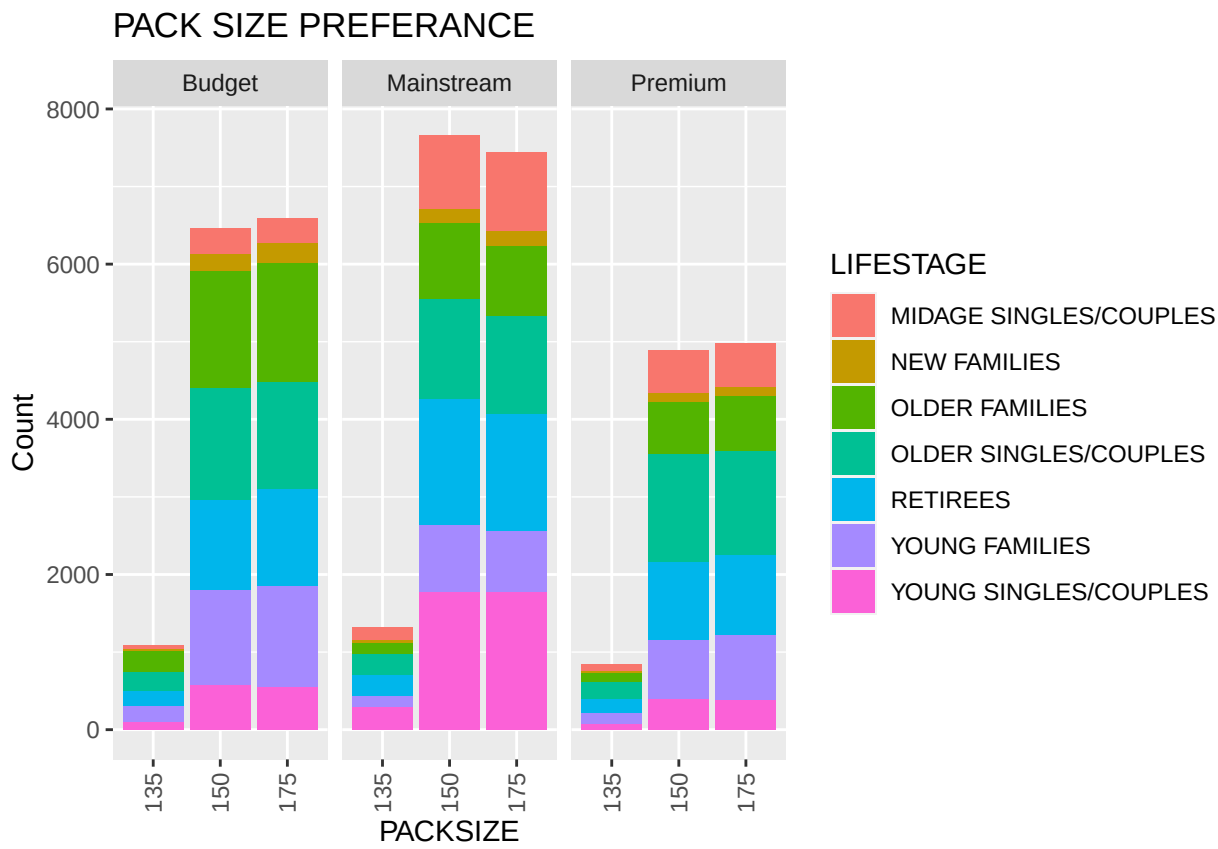

```
##Kettle is the second most expensive Brand. Yet, customers are willing to an average $4.94 for it.

####We can check what packsizes customer segments prefer
data %>% group_by(PACK_SIZE, LIFESTAGE, PREMIUM_CUSTOMER, Brand) %>% summarise(num = n()) %>%
  ggplot(aes(x = as.character(PACK_SIZE), y = num, fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "PACKSIZE", y = "Count", title = "PACK SIZE PREFERANCE")
```

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




```
##Most customers prefer the 175g packsize. They also like the 150g, 134g, and 110g pack sizes.


##Let's zoom in and see what "Kettle" packsizes are prefered
data %>% group_by(PACK_SIZE, LIFESTAGE, PREMIUM_CUSTOMER, Brand) %>%
  filter(Brand == "Kettle") %>% summarise(num = n()) %>%
  ggplot(aes(x = as.character(PACK_SIZE), y = num, fill = LIFESTAGE)) +
  geom_bar(stat ="identity") + facet_grid(~PREMIUM_CUSTOMER) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  labs(x = "PACKSIZE", y = "Count", title = "PACK SIZE PREFERANCE")
```

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




```
##The customer's favourite chips brands "Kettle" has only the 175g, 150g, and 135g Packsizes available.
##Futhermore, most mainsteam prefer the 150g packsize, while others prefer 175.
##The difference is however not very significant.

####CONCLUSION
#Let's recap all that we've found:
#Sales have mainly been due to Budget - older families, Mainstream - young singles/couples,
#and Mainstream retirees shoppers.
##We found that the high spend in chips for mainstream young singles/couples
#and retirees is due to there being more of them than other buyers.
#Mainstream, midage and young singles and couples are also more likely to pay more per packet of chips.
#This is indicative of impulse buying behaviour.
##We've also found that Mainstream young singles and couples are 23% more likely to purchase Tyrrells c
#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.
```