quantium

kiran jangili

2024-08-07

```
library(data.table)
library(ggplot2)
library(ggmosaic)
library(readr)
```

```
filePath<- ""
transactionData<-fread(paste0(filePath,"QVI_transaction_data.csv"))
customerData<-fread(paste0(filePath,"QVI_purchase_behaviour.csv"))
```

```
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>
```

```
head(transactionData)
```

```
##     DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
##    <int>     <int>          <int>  <int>    <int>
## 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
##                                       <char>    <int>     <num>
## 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
```

```
summary(transactionData)
```

```
##       DATE          STORE_NBR     LYLTY_CARD_NBR        TXN_ID
##  Min.   :43282   Min.   :  1.0   Min.   :   1000   Min.   :       1
##  1st Qu.:43373   1st Qu.: 70.0   1st Qu.:  70021   1st Qu.:  67602
##  Median :43464   Median :130.0   Median : 130358   Median : 135138
##  Mean   :43464   Mean   :135.1   Mean   : 135550   Mean   : 135158
##  3rd Qu.:43555   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202701
##  Max.   :43646   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR        PROD_NAME              PROD_QTY         TOT_SALES
##  Min.   :  1.00   Length:264836      Min.   :  1.000   Min.   :  1.500
##  1st Qu.: 28.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.400
##  Median : 56.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.58                      Mean   :  1.907   Mean   :  7.304
##  3rd Qu.: 85.00                      3rd Qu.:  2.000   3rd Qu.:  9.200
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
```

convert DATE to date format

```
transactionData$DATE<- as.Date(transactionData$DATE, origin="1899-12-30")

summary(transactionData$PROD_NAME)
```

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

```
productWords<- data.table(words=unlist(strsplit(transactionData$PROD_NAME,"
")))
```

removing digits

```
productWords<-productWords[!grepl("\\d", words),]
```

removing special characters

```
productWords<-productWords[!grepl("[^[:alnum:] ]",words), ]
```

most common words

```
wordFreq<-productWords[, .N,by = words][order(-N)]
head(wordFreq,10)
```

```
##        words      N
##       <char>  <int>
##  1:          504838
##  2:    Chips  49770
##  3:   Kettle  41288
##  4:   Smiths  28860
##  5:     Salt  27976
##  6:   Cheese  27890
##  7: Pringles  25102
##  8:  Doritos  24962
##  9:  Crinkle  23960
## 10:     Corn  22063
```

removing salsa products

```
transactionData[, SALSA:= grepl("salsa",tolower(PROD_NAME))]
transactionData<- transactionData[SALSA==FALSE, ]
transactionData[, SALSA := NULL]
```

checking for outliers

```
outliers<-transactionData[PROD_QTY==200]
print(outliers)

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

see if the customer has another transaction

```
customer_id<- outliers$LYLTY_CARD_NBR[1]
customer_transactions<-transactionData[LYLTY_CARD_NBR==customer_id]
print(customer_transactions)

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

finding out the customer based on the layality card number

```
transactionData<- transactionData[LYLTY_CARD_NBR != customer_id]
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
```

count the number of transaction by date

```
transaction_by_date<-transactionData[, .N, by=DATE]
print(transaction_by_date)
```

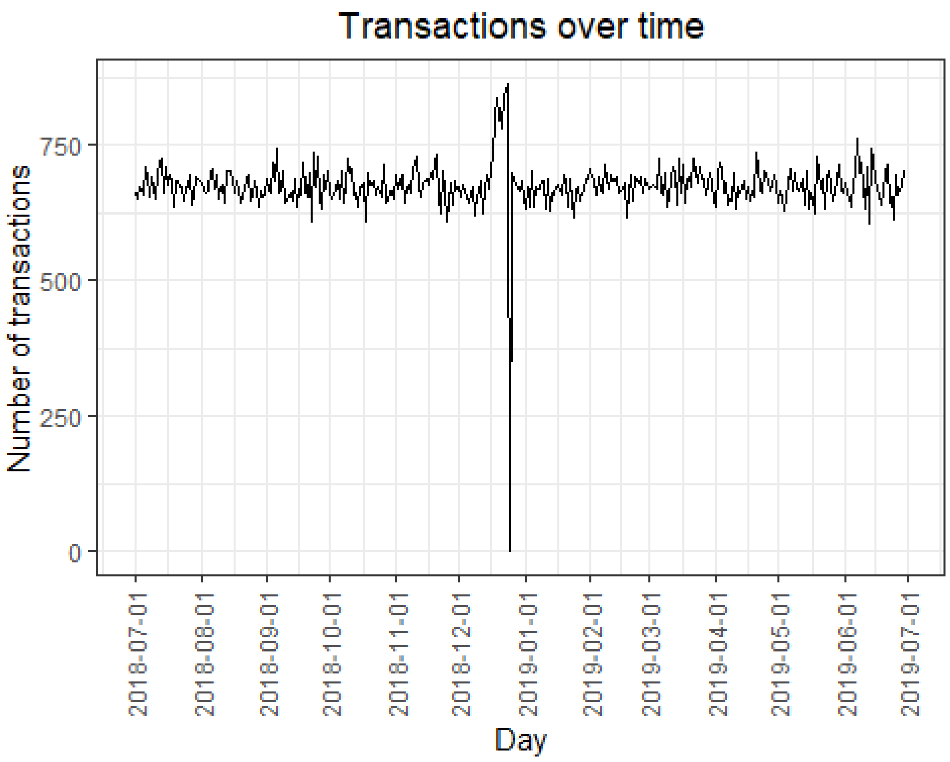
```
##            DATE     N
##          <Date> <int>
##   1: 2018-10-17   682
##   2: 2019-05-14   705
##   3: 2019-05-20   707
##   4: 2018-08-17   663
##   5: 2018-08-18   683
##  ---
## 360: 2018-12-08   622
## 361: 2019-01-30   689
## 362: 2019-02-09   671
## 363: 2018-08-31   658
## 364: 2019-02-12   684
```

create a sequence of dates and join this count of transaction by date

```
all_dates<-data.table(DATE=seq.Date(as.Date("2018-07-01"), as.Date("2019-06-30"), by="day"))
transaction_by_date<-merge(all_dates,transaction_by_date, by="DATE" , all.x = TRUE)
transaction_by_date[is.na(N),N :=0]

theme_set(theme_bw())
theme_update(plot.title=element_text(hjust = 0.5))
```

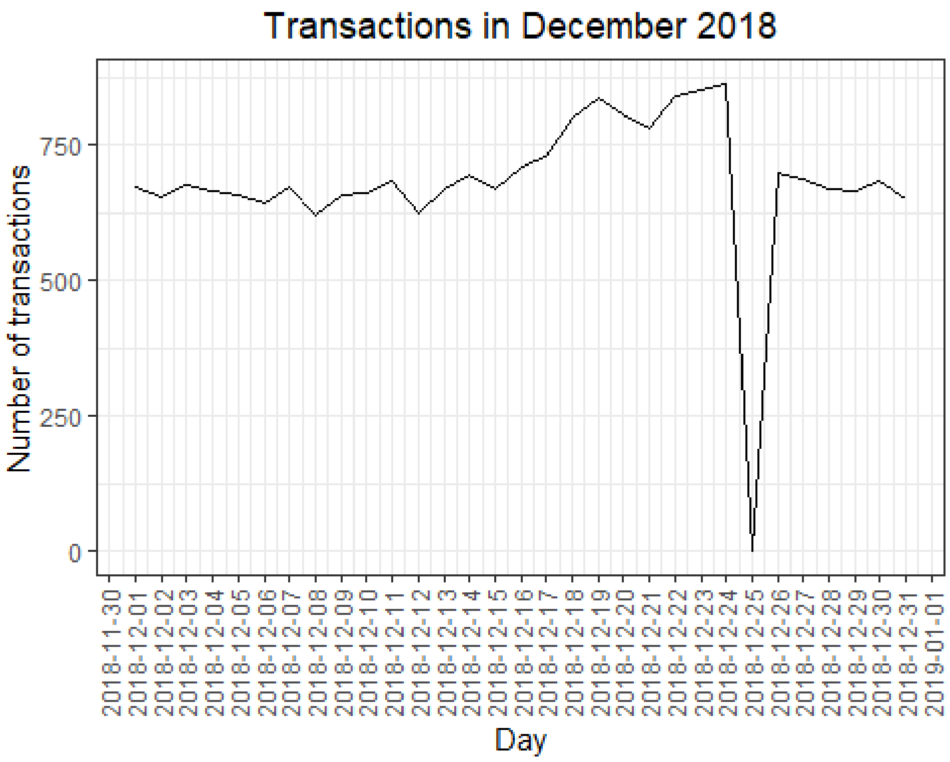
plotting

```
ggplot(transaction_by_date,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))
```



Recreate the chart above zoomed in to the relevant dates

```
december_transactions<-transaction_by_date[DATE>="2018-12-01" & DATE<= "2018-
12-31"]

ggplot(december_transactions,aes(x=DATE,y=N))+
    geom_line()+
            labs(x="Day",y="Number of transactions", title = "Transactions in
December 2018")+
                scale_x_date(breaks = "1 day")+
                                  theme(axis.text.x = element_text(angle =
90,vjust = 0.5))
```



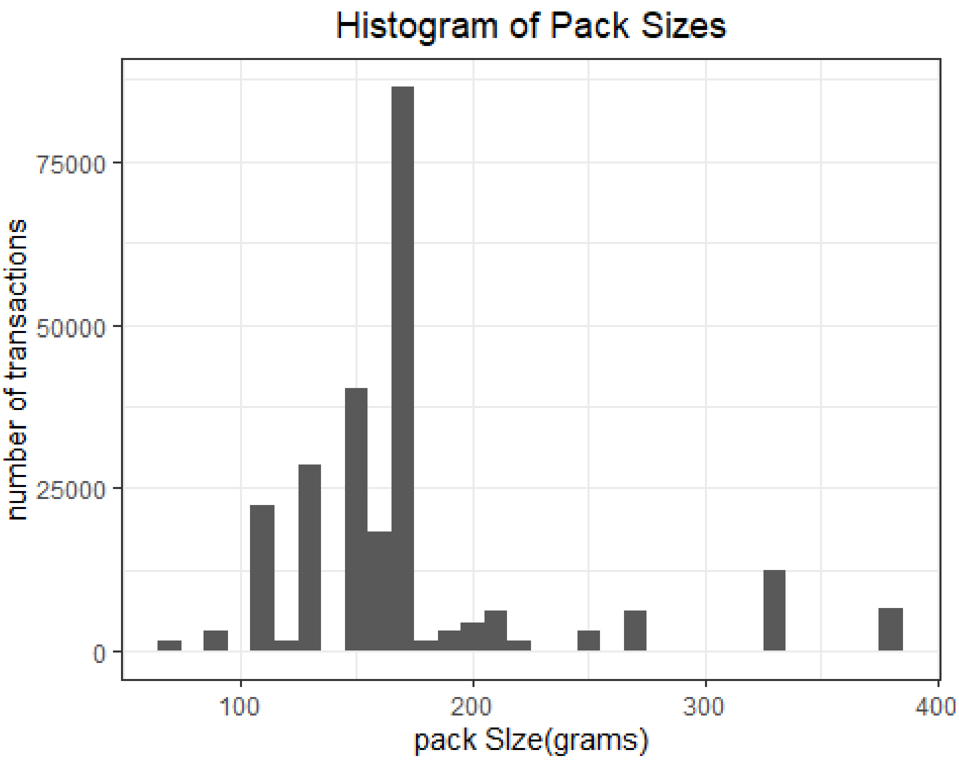
pack size

```
transactionData[,PACK_SIZE:= parse_number(PROD_NAME)]
transactionData[, .N,PACK_SIZE] [order(PACK_SIZE)]
```

```
##     PACK_SIZE     N
##         <num> <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
##     PACK_SIZE     N
```

plot a histogram showing the number of transactions by pack size

```
ggplot(transactionData, aes(x=PACK_SIZE))+
      geom_histogram(binwidth = 10)+
                      labs(x="pack SIze(grams)", y="number of transactions" ,
title = "Histogram of Pack Sizes")
```



creating a cloumn which contains the brand of the product , by extracting it from the product name

```
transactionData[, BRAND:=toupper(sub(" .*", "", PROD_NAME))]
print(unique(transactionData$BRAND))
```

```
##  [1] "NATURAL"    "CCS"        "SMITHS"     "KETTLE"     "GRAIN"
##  [6] "DORITOS"    "TWISTIES"   "WW"         "THINS"      "BURGER"
## [11] "NCC"        "CHEEZELS"   "INFZNS"     "RED"        "PRINGLES"
## [16] "DORITO"     "INFUZIONS"  "SMITH"      "GRNWVES"    "TYRRELLS"
## [21] "COBS"       "FRENCH"     "RRD"        "TOSTITOS"   "CHEETOS"
## [26] "WOOLWORTHS" "SNBTS"      "SUNBITES"
```

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=="DORITO",BRAND:="DORITOS"]
transactionData[BRAND=="NCC",BRAND:="NATURAL"]
transactionData[BRAND=="GRAIN",BRAND:="GRNWVES"]
```

```
transactionData[BRAND=="CHEEZEL",BRAND:="CHEEZELS"]
print(unique(transactionData$BRAND))
```

```
##  [1] "NATURAL"    "CCS"        "SMITHS"     "KETTLE"     "GRNWVES"   
##  [6] "DORITOS"    "TWISTIES"   "WOOLWORTHS" "THINS"      "BURGER"    
## [11] "CHEEZELS"   "INFUZIONS"  "RRD"        "PRINGLES"   "TYRRELLS"  
## [16] "COBS"       "FRENCH"     "TOSTITOS"   "CHEETOS"    "SUNBITES"
```

Examining customer data

```
str(customerData)
```

```
## Classes 'data.table' and 'data.frame':   72637 obs. of  3 variables:
##  $ LYLTY_CARD_NBR  : int  1000 1002 1003 1004 1005 1007 1009 1010 1011
1012 ...
##  $ LIFESTAGE       : chr  "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES"
"YOUNG FAMILIES" "OLDER SINGLES/COUPLES" ...
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
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
```

Merge transaction data to customer data

```
data<-merge(transactionData,customerData,all.x = TRUE)
head(data)
```

```
## Key: <LYLTY_CARD_NBR>
##    LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR
##             <int>     <Date>     <int>  <int>    <int>
## 1:           1000 2018-10-17         1      1        5
## 2:           1002 2018-09-16         1      2       58
## 3:           1003 2019-03-07         1      3       52
## 4:           1003 2019-03-08         1      4      106
## 5:           1004 2018-11-02         1      5       96
## 6:           1005 2018-12-28         1      6       86
##                                   PROD_NAME PROD_QTY TOT_SALES PACK_SIZE
##                                      <char>    <int>     <num>     <num>
## 1: Natural Chip        Compny SeaSalt175g        2       6.0       175
## 2:  Red Rock Deli Chikn&Garlic Aioli 150g        1       2.7       150
## 3:  Grain Waves Sour    Cream&Chives 210G        1       3.6       210
## 4: Natural ChipCo      Hony Soy Chckn175g        1       3.0       175
## 5:         WW Original Stacked Chips 160g        1       1.9       160
## 6:                     Cheetos Puffs 165g        1       2.8       165
##         BRAND              LIFESTAGE PREMIUM_CUSTOMER
##        <char>                <char>           <char>
## 1:    NATURAL  YOUNG SINGLES/COUPLES          Premium
## 2:        RRD  YOUNG SINGLES/COUPLES       Mainstream
## 3:    GRNWVES         YOUNG FAMILIES           Budget
## 4:    NATURAL         YOUNG FAMILIES           Budget
## 5: WOOLWORTHS  OLDER SINGLES/COUPLES       Mainstream
## 6:    CHEETOS MIDAGE SINGLES/COUPLES       Mainstream

View(data)
```

see if any transactions did not have a matched customer

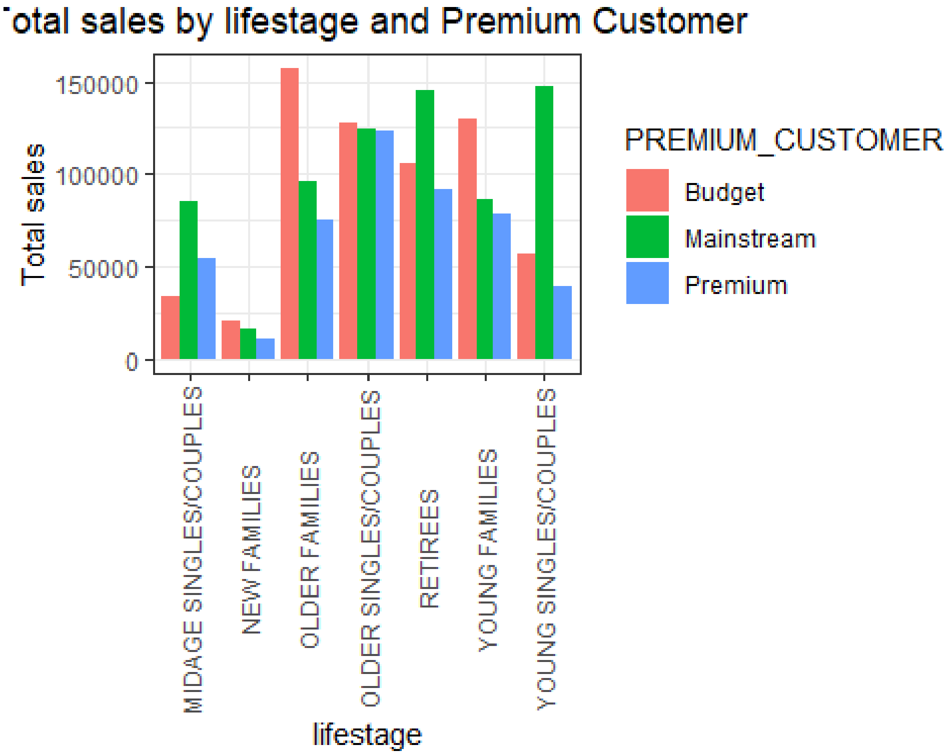
```
missing_customer<-transactionData[!LYLTY_CARD_NBR %in%
customerData$LYLTY_CARD_NBR]
print(missing_customer)

## Empty data.table (0 rows and 10 cols):
DATE,STORE_NBR,LYLTY_CARD_NBR,TXN_ID,PROD_NBR,PROD_NAME...
```

csv format

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

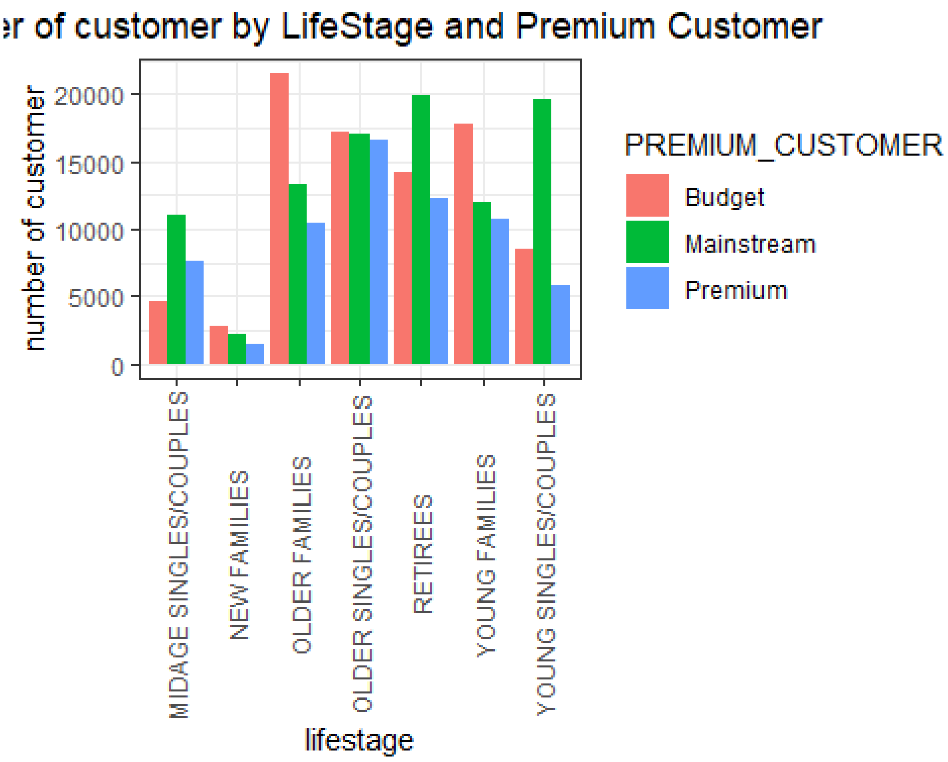
calculate the summary of slaes by those dimensions and create a plot

```
total_sales<-data[, .(TOTAL_SALES=sum(TOT_SALES)), by=.(LIFESTAGE,
PREMIUM_CUSTOMER)]
ggplot(total_sales,aes(x=LIFESTAGE,y=TOTAL_SALES,fill=PREMIUM_CUSTOMER))+
      geom_bar(stat = "identity",position = "dodge")+
        labs(x="lifestage",y="Total sales", title = "Total sales by lifestage
and Premium Customer")+
          theme(axis.text.x = element_text(angle = 90,vjust = 0.5))
```



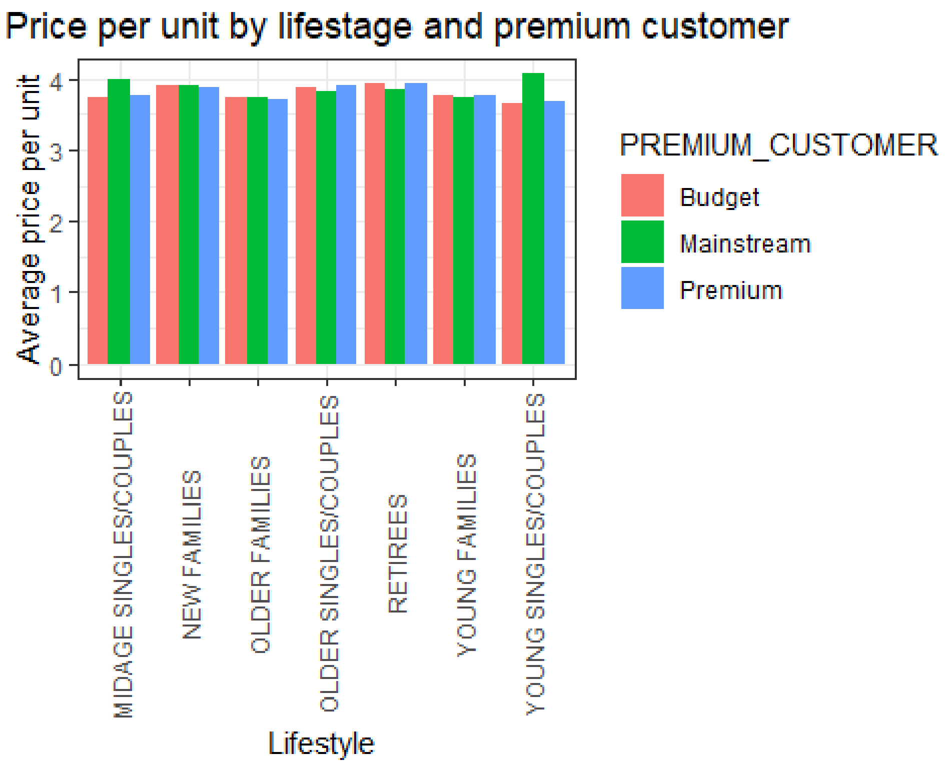
calculate the summary of number of customer by those dimension and create a plot

```
customer_count<-data[,.N, by=.(LIFESTAGE, PREMIUM_CUSTOMER)]
ggplot(customer_count,aes(x=LIFESTAGE, y=N, fill=PREMIUM_CUSTOMER))+
    geom_bar(stat="identity", position='dodge')+
  labs(x="lifestage",y="number of customer", title = "Number of customer by
LifeStage and Premium Customer")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



calculate and plot the average price per unit sold (average sale price) by those two customer dimension

```
avg_price<- data[, .(AVG_PRICE=mean(TOT_SALES/PROD_QTY)), by=.(LIFESTAGE,
PREMIUM_CUSTOMER)]
ggplot(avg_price,aes(x=LIFESTAGE, y=AVG_PRICE, fill = PREMIUM_CUSTOMER))+
    geom_bar(stat = "identity", position = "dodge")+
        labs(x="Lifestyle ", y="Average price per unit", title = "Average
Price per unit by lifestage and premium customer")+
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))+
      theme(axis.title.x = element_text(vjust = 0.5))
```



Filter the data for mainstream and premium/budget young and midage singles and couples

```
mainstream_young_midage <- data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES",
"MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER == "Mainstream"]
premium_budget_young_midage <- data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES",
"MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER %in% c("Premium", "Budget")]
```

Conduct the t-test on the unit price

```
t_test_result <- t.test(mainstream_young_midage$TOT_SALES /
mainstream_young_midage$PROD_QTY,
                        premium_budget_young_midage$TOT_SALES /
premium_budget_young_midage$PROD_QTY)
```

Print the t-test result

```
print(t_test_result)

##
##  Welch Two Sample t-test
##
## data:  mainstream_young_midage$TOT_SALES/mainstream_young_midage$PROD_QTY
and
premium_budget_young_midage$TOT_SALES/premium_budget_young_midage$PROD_QTY
## t = 37.624, df = 54791, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3159319 0.3506572
## sample estimates:
```

```
## mean of x mean of y
##  4.039786  3.706491
```

Interpret the p-value from the t-test

```
p_value <- t_test_result$p.value
if (p_value < 0.05) {
  cat("The t-test results in a p-value of", p_value, ", 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.\n")
} else {
  cat("The t-test results in a p-value of", p_value, ", i.e. the unit price
for mainstream, young and mid-age singles and couples ARE NOT significantly
higher than that of budget or premium, young and midage singles and
couples.\n")
}

## The t-test results in a p-value of 6.967354e-306 , 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 Mainstream, young singles/couple

```
mainstream_young_singles_couples <- data[LIFESTAGE == "YOUNG SINGLES/COUPLES"
& PREMIUM_CUSTOMER == "Mainstream"]
```

Calculate the frequency of each brand bought by this segment

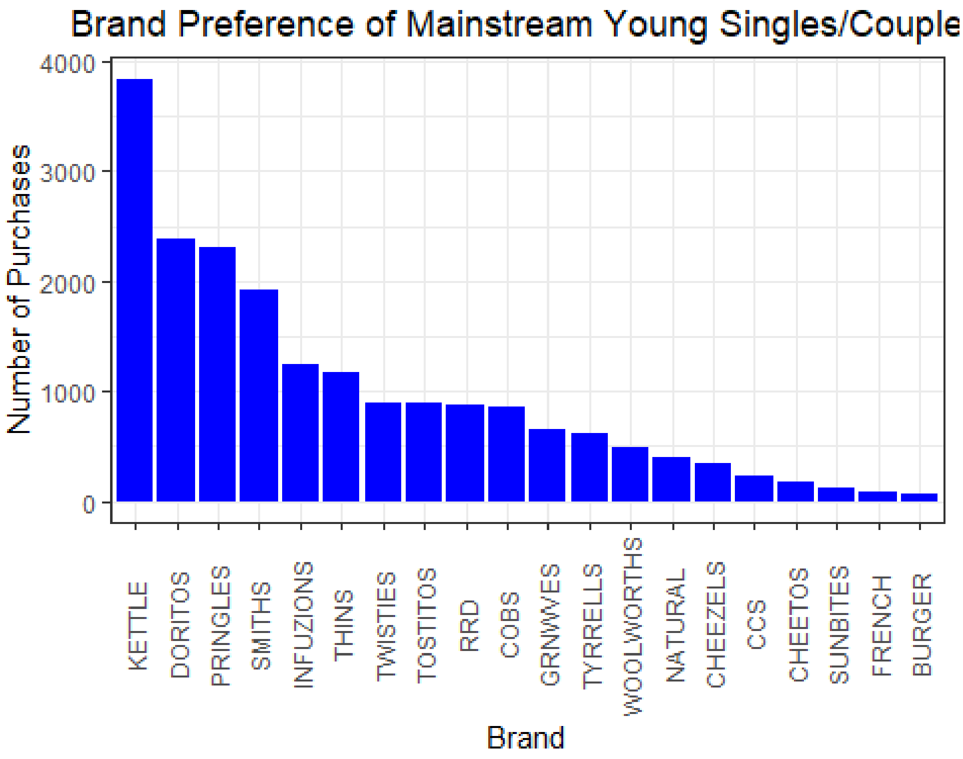
```
brand_preference <- mainstream_young_singles_couples[, .N, by = .(BRAND)]
brand_preference <- brand_preference[order(-N)]
```

Print the top brands

```
print(head(brand_preference, 10))

##         BRAND     N
##        <char> <int>
##  1:    KETTLE  3844
##  2:   DORITOS  2379
##  3:  PRINGLES  2315
##  4:    SMITHS  1921
##  5: INFUZIONS  1250
##  6:     THINS  1166
##  7:  TWISTIES   900
##  8:  TOSTITOS   890
##  9:       RRD   875
## 10:      COBS   864
```

Plot the brand preference

```
ggplot(brand_preference, aes(x = reorder(BRAND, -N), y = N)) +
  geom_bar(stat = "identity", fill = "blue") +
  labs(x = "Brand", y = "Number of Purchases", title = "Brand Preference of
Mainstream Young Singles/Couples") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5)) +
  theme(plot.title = element_text(hjust = 0.5))
```

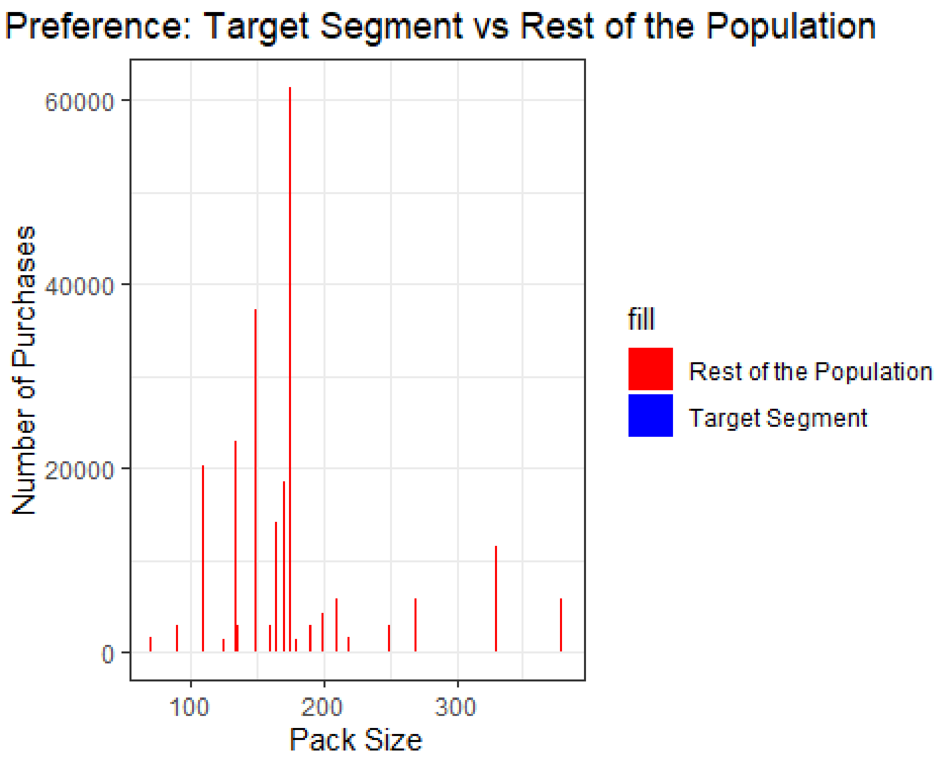


Compare pack size preference between target segment and rest of the population

```
pack_size_preference_target <- mainstream_young_singles_couples[, .N, by =
.(PACK_SIZE)]
pack_size_preference_rest <- data[LIFESTAGE != "YOUNG SINGLES/COUPLES" |
PREMIUM_CUSTOMER != "Mainstream", .N, by = .(PACK_SIZE)]
```

Plot pack size preference

```
ggplot() +
  geom_bar(data = pack_size_preference_target, aes(x = PACK_SIZE, y = N, fill
= "Target Segment"), stat = "identity", position = "dodge") +
  geom_bar(data = pack_size_preference_rest, aes(x = PACK_SIZE, y = N, fill =
"Rest of the Population"), stat = "identity", position = "dodge") +
  labs(x = "Pack Size", y = "Number of Purchases", title = "Pack Size
Preference: Target Segment vs Rest of the Population") +
  scale_fill_manual(values = c("Target Segment" = "blue", "Rest of the
Population" = "red")) +
  theme(plot.title = element_text(hjust = 0.5))
```


Preference: Target Segment vs Rest of the Population