Quantium Virtual Internship - Retail Strategy and Analytics - Task 1

Yussuf Ali

2024-09-07

```
# Install required packages if not already installed
#install.packages("data.table")
#install.packages("ggplot2")
#install.packages("readr")
#install.packages("readxl")

# Load required libraries
library(data.table)
library(ggplot2)
library(readr)
library(readxl)

getwd()
setwd("C:/Users/USER/Desktop/Python program/Data_Analytics_internship/")


filePath <- "C:/Users/USER/Desktop/Python program/Data_Analytics_internship/"
```

```
# Load the data using fread with the correct file path
transactionData <- fread(paste0(filePath,"QVI_transaction_data.csv"))
# Check if the data is loaded successfully
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" "Smith
##  $ 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, n=10)
```

```
##      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
##  7: 43601         4           4149   3333       16
##  8: 43601         4           4196   3539       24
##  9: 43332         5           5026   4525       42
## 10: 43330         7           7150   6900       52
##                                     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
##  7: Smiths Crinkle Chips Salt & Vinegar 330g        1       5.7
##  8:    Grain Waves         Sweet Chilli 210g        1       3.6
##  9:  Doritos Corn Chip Mexican Jalapeno 150g        1       3.9
## 10:    Grain Waves Sour    Cream&Chives 210G        2       7.2
```

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

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" "OLD
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```r
head(customerData, n=10)
```

```
##     LYLTY_CARD_NBR              LIFESTAGE PREMIUM_CUSTOMER
##              <int>                 <char>           <char>
##  1:           1000  YOUNG SINGLES/COUPLES          Premium
##  2:           1002  YOUNG SINGLES/COUPLES       Mainstream
##  3:           1003         YOUNG FAMILIES           Budget
##  4:           1004  OLDER SINGLES/COUPLES       Mainstream
##  5:           1005 MIDAGE SINGLES/COUPLES       Mainstream
##  6:           1007  YOUNG SINGLES/COUPLES           Budget
##  7:           1009           NEW FAMILIES          Premium
##  8:           1010  YOUNG SINGLES/COUPLES       Mainstream
##  9:           1011  OLDER SINGLES/COUPLES       Mainstream
## 10:           1012         OLDER FAMILIES       Mainstream
```

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

```r
# Check the conversion
str(transactionData$DATE)
```

```
##   Date[1:264836], format: "2018-10-17" "2019-05-14" "2019-05-20" "2018-08-
17" "2018-08-18" ...
```

```r
# Check for missing values in each column of the transaction data
missing_values <- colSums(is.na(transactionData))
print(missing_values)
```

```
##           DATE      STORE_NBR LYLTY_CARD_NBR         TXN_ID       PROD_NBR
##              0              0              0              0              0
##      PROD_NAME       PROD_QTY      TOT_SALES
##              0              0              0
```

```r
# Generate a summary of the PROD_NAME column
summary(transactionData$PROD_NAME)
```

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

```r
# View the unique product names to identify non-chip entries
unique(transactionData$PROD_NAME)
```

```
##   [1] "Natural Chip        Compny SeaSalt175g"
##   [2] "CCs Nacho Cheese    175g"
##   [3] "Smiths Crinkle Cut  Chips Chicken 170g"
##   [4] "Smiths Chip Thinly  S/Cream&Onion 175g"
##   [5] "Kettle Tortilla ChpsHny&Jlpno Chili 150g"
##   [6] "Old El Paso Salsa   Dip Tomato Mild 300g"
##   [7] "Smiths Crinkle Chips Salt & Vinegar 330g"
##   [8] "Grain Waves         Sweet Chilli 210g"
##   [9] "Doritos Corn Chip Mexican Jalapeno 150g"
##  [10] "Grain Waves Sour    Cream&Chives 210G"
##  [11] "Kettle Sensations   Siracha Lime 150g"
##  [12] "Twisties Cheese     270g"
##  [13] "WW Crinkle Cut      Chicken 175g"
##  [14] "Thins Chips Light&  Tangy 175g"
##  [15] "CCs Original 175g"
##  [16] "Burger Rings 220g"
##  [17] "NCC Sour Cream &    Garden Chives 175g"
##  [18] "Doritos Corn Chip Southern Chicken 150g"
##  [19] "Cheezels Cheese Box 125g"
##  [20] "Smiths Crinkle      Original 330g"
##  [21] "Infzns Crn Crnchers Tangy Gcamole 110g"
##  [22] "Kettle Sea Salt     And Vinegar 175g"
##  [23] "Smiths Chip Thinly  Cut Original 175g"
##  [24] "Kettle Original 175g"
##  [25] "Red Rock Deli Thai  Chilli&Lime 150g"
##  [26] "Pringles Sthrn FriedChicken 134g"
```

```
##  [27] "Pringles Sweet&Spcy BBQ 134g"
##  [28] "Red Rock Deli SR    Salsa & Mzzrlla 150g"
##  [29] "Thins Chips         Originl saltd 175g"
##  [30] "Red Rock Deli Sp    Salt & Truffle 150G"
##  [31] "Smiths Thinly       Swt Chli&S/Cream175G"
##  [32] "Kettle Chilli 175g"
##  [33] "Doritos Mexicana    170g"
##  [34] "Smiths Crinkle Cut  French OnionDip 150g"
##  [35] "Natural ChipCo      Hony Soy Chckn175g"
##  [36] "Dorito Corn Chp     Supreme 380g"
##  [37] "Twisties Chicken270g"
##  [38] "Smiths Thinly Cut   Roast Chicken 175g"
##  [39] "Smiths Crinkle Cut  Tomato Salsa 150g"
##  [40] "Kettle Mozzarella   Basil & Pesto 175g"
##  [41] "Infuzions Thai SweetChili PotatoMix 110g"
##  [42] "Kettle Sensations   Camembert & Fig 150g"
##  [43] "Smith Crinkle Cut   Mac N Cheese 150g"
##  [44] "Kettle Honey Soy    Chicken 175g"
##  [45] "Thins Chips Seasonedchicken 175g"
##  [46] "Smiths Crinkle Cut  Salt & Vinegar 170g"
##  [47] "Infuzions BBQ Rib   Prawn Crackers 110g"
##  [48] "GrnWves Plus Btroot & Chilli Jam 180g"
##  [49] "Tyrrells Crisps     Lightly Salted 165g"
##  [50] "Kettle Sweet Chilli And Sour Cream 175g"
##  [51] "Doritos Salsa       Medium 300g"
##  [52] "Kettle 135g Swt Pot Sea Salt"
##  [53] "Pringles SourCream  Onion 134g"
##  [54] "Doritos Corn Chips  Original 170g"
##  [55] "Twisties Cheese     Burger 250g"
##  [56] "Old El Paso Salsa   Dip Chnky Tom Ht300g"
##  [57] "Cobs Popd Swt/Chlli &Sr/Cream Chips 110g"
##  [58] "Woolworths Mild     Salsa 300g"
##  [59] "Natural Chip Co     Tmato Hrb&Spce 175g"
##  [60] "Smiths Crinkle Cut  Chips Original 170g"
##  [61] "Cobs Popd Sea Salt  Chips 110g"
##  [62] "Smiths Crinkle Cut  Chips Chs&Onion170g"
##  [63] "French Fries Potato Chips 175g"
##  [64] "Old El Paso Salsa   Dip Tomato Med 300g"
##  [65] "Doritos Corn Chips  Cheese Supreme 170g"
##  [66] "Pringles Original   Crisps 134g"
##  [67] "RRD Chilli&         Coconut 150g"
##  [68] "WW Original Corn    Chips 200g"
##  [69] "Thins Potato Chips  Hot & Spicy 175g"
##  [70] "Cobs Popd Sour Crm  &Chives Chips 110g"
##  [71] "Smiths Crnkle Chip  Orgnl Big Bag 380g"
##  [72] "Doritos Corn Chips  Nacho Cheese 170g"
##  [73] "Kettle Sensations   BBQ&Maple 150g"
##  [74] "WW D/Style Chip     Sea Salt 200g"
##  [75] "Pringles Chicken    Salt Crips 134g"
##  [76] "WW Original Stacked Chips 160g"
##  [77] "Smiths Chip Thinly  CutSalt/Vinegr175g"
##  [78] "Cheezels Cheese 330g"
##  [79] "Tostitos Lightly    Salted 175g"
##  [80] "Thins Chips Salt &  Vinegar 175g"
```

```
##  [81] "Smiths Crinkle Cut  Chips Barbecue 170g"
##  [82] "Cheetos Puffs 165g"
##  [83] "RRD Sweet Chilli &  Sour Cream 165g"
##  [84] "WW Crinkle Cut      Original 175g"
##  [85] "Tostitos Splash Of  Lime 175g"
##  [86] "Woolworths Medium   Salsa 300g"
##  [87] "Kettle Tortilla ChpsBtroot&Ricotta 150g"
##  [88] "CCs Tasty Cheese    175g"
##  [89] "Woolworths Cheese   Rings 190g"
##  [90] "Tostitos Smoked     Chipotle 175g"
##  [91] "Pringles Barbeque   134g"
##  [92] "WW Supreme Cheese   Corn Chips 200g"
##  [93] "Pringles Mystery    Flavour 134g"
##  [94] "Tyrrells Crisps     Ched & Chives 165g"
##  [95] "Snbts Whlgrn Crisps Cheddr&Mstrd 90g"
##  [96] "Cheetos Chs & Bacon Balls 190g"
##  [97] "Pringles Slt Vingar 134g"
##  [98] "Infuzions SourCream&Herbs Veg Strws 110g"
##  [99] "Kettle Tortilla ChpsFeta&Garlic 150g"
## [100] "Infuzions Mango     Chutny Papadums 70g"
## [101] "RRD Steak &         Chimuchurri 150g"
## [102] "RRD Honey Soy       Chicken 165g"
## [103] "Sunbites Whlegrn    Crisps Frch/Onin 90g"
## [104] "RRD Salt & Vinegar  165g"
## [105] "Doritos Cheese      Supreme 330g"
## [106] "Smiths Crinkle Cut  Snag&Sauce 150g"
## [107] "WW Sour Cream &OnionStacked Chips 160g"
## [108] "RRD Lime & Pepper   165g"
## [109] "Natural ChipCo Sea  Salt & Vinegr 175g"
## [110] "Red Rock Deli Chikn&Garlic Aioli 150g"
## [111] "RRD SR Slow Rst     Pork Belly 150g"
## [112] "RRD Pc Sea Salt     165g"
## [113] "Smith Crinkle Cut   Bolognese 150g"
## [114] "Doritos Salsa Mild  300g"
```

```
# Load necessary library for string manipulation
#install.packages("stringr")
library(stringr)

# Split product names into individual words
productWords <- data.table(unlist(strsplit(unique(transactionData$PROD_NAME), "
 ↪  ")))
setnames(productWords, 'words')

# View the first few words extracted
head(productWords)
```

```
##      words
##     <char>
## 1: Natural
## 2:    Chip
## 3:
## 4:
```

```
## 5:
## 6:
```

```
# Split product names into individual words and create a data table
productWords <- data.table(unlist(strsplit(unique(transactionData$PROD_NAME), "
 ↪  ")))
setnames(productWords, 'words')

# Remove empty strings and spaces
productWords <- productWords[words != ""]

# Remove words with digits and special characters
productWords <- productWords[!grepl("\\d", words)]
productWords <- productWords[!grepl("[^a-zA-Z]", words)]

# Count the frequency of each word
wordFrequency <- productWords[, .N, by = words][order(-N)]

# Display the most common words
print(wordFrequency)
```

```
##          words     N
##         <char> <int>
##   1:     Chips    21
##   2:    Smiths    16
##   3:   Crinkle    14
##   4:       Cut    14
##   5:    Kettle    13
##  ---
## 164:       Rst     1
## 165:      Pork     1
## 166:     Belly     1
## 167:        Pc     1
## 168: Bolognese     1
```

```
transactionData <- as.data.table(transactionData)

# Remove salsa products from the dataset
transactionData[, SALSA := grepl("salsa", tolower(PROD_NAME))]

# Keep only entries where SALSA is FALSE
transactionData <- transactionData[SALSA == FALSE, ][, SALSA := NULL]

# Verify the removal of salsa products by checking unique product names
unique(transactionData$PROD_NAME)
```

```
##   [1] "Natural Chip        Compny SeaSalt175g"
##   [2] "CCs Nacho Cheese    175g"
##   [3] "Smiths Crinkle Cut  Chips Chicken 170g"
##   [4] "Smiths Chip Thinly  S/Cream&Onion 175g"
##   [5] "Kettle Tortilla ChpsHny&Jlpno Chili 150g"
##   [6] "Smiths Crinkle Chips Salt & Vinegar 330g"
```

```
##   [7] "Grain Waves         Sweet Chilli 210g"
##   [8] "Doritos Corn Chip Mexican Jalapeno 150g"
##   [9] "Grain Waves Sour    Cream&Chives 210G"
##  [10] "Kettle Sensations   Siracha Lime 150g"
##  [11] "Twisties Cheese     270g"
##  [12] "WW Crinkle Cut      Chicken 175g"
##  [13] "Thins Chips Light&  Tangy 175g"
##  [14] "CCs Original 175g"
##  [15] "Burger Rings 220g"
##  [16] "NCC Sour Cream &    Garden Chives 175g"
##  [17] "Doritos Corn Chip Southern Chicken 150g"
##  [18] "Cheezels Cheese Box 125g"
##  [19] "Smiths Crinkle      Original 330g"
##  [20] "Infzns Crn Crnchers Tangy Gcamole 110g"
##  [21] "Kettle Sea Salt     And Vinegar 175g"
##  [22] "Smiths Chip Thinly  Cut Original 175g"
##  [23] "Kettle Original 175g"
##  [24] "Red Rock Deli Thai  Chilli&Lime 150g"
##  [25] "Pringles Sthrn FriedChicken 134g"
##  [26] "Pringles Sweet&Spcy BBQ 134g"
##  [27] "Thins Chips         Originl saltd 175g"
##  [28] "Red Rock Deli Sp    Salt & Truffle 150G"
##  [29] "Smiths Thinly       Swt Chli&S/Cream175G"
##  [30] "Kettle Chilli 175g"
##  [31] "Doritos Mexicana    170g"
##  [32] "Smiths Crinkle Cut  French OnionDip 150g"
##  [33] "Natural ChipCo      Hony Soy Chckn175g"
##  [34] "Dorito Corn Chp     Supreme 380g"
##  [35] "Twisties Chicken270g"
##  [36] "Smiths Thinly Cut   Roast Chicken 175g"
##  [37] "Kettle Mozzarella   Basil & Pesto 175g"
##  [38] "Infuzions Thai SweetChili PotatoMix 110g"
##  [39] "Kettle Sensations   Camembert & Fig 150g"
##  [40] "Smith Crinkle Cut   Mac N Cheese 150g"
##  [41] "Kettle Honey Soy    Chicken 175g"
##  [42] "Thins Chips Seasonedchicken 175g"
##  [43] "Smiths Crinkle Cut  Salt & Vinegar 170g"
##  [44] "Infuzions BBQ Rib   Prawn Crackers 110g"
##  [45] "GrnWves Plus Btroot & Chilli Jam 180g"
##  [46] "Tyrrells Crisps     Lightly Salted 165g"
##  [47] "Kettle Sweet Chilli And Sour Cream 175g"
##  [48] "Kettle 135g Swt Pot Sea Salt"
##  [49] "Pringles SourCream  Onion 134g"
##  [50] "Doritos Corn Chips  Original 170g"
##  [51] "Twisties Cheese     Burger 250g"
##  [52] "Cobs Popd Swt/Chlli &Sr/Cream Chips 110g"
##  [53] "Natural Chip Co     Tmato Hrb&Spce 175g"
##  [54] "Smiths Crinkle Cut  Chips Original 170g"
##  [55] "Cobs Popd Sea Salt  Chips 110g"
##  [56] "Smiths Crinkle Cut  Chips Chs&Onion170g"
##  [57] "French Fries Potato Chips 175g"
##  [58] "Doritos Corn Chips  Cheese Supreme 170g"
##  [59] "Pringles Original   Crisps 134g"
##  [60] "RRD Chilli&         Coconut 150g"
```

```
##  [61] "WW Original Corn    Chips 200g"
##  [62] "Thins Potato Chips  Hot & Spicy 175g"
##  [63] "Cobs Popd Sour Crm  &Chives Chips 110g"
##  [64] "Smiths Crnkle Chip  Orgnl Big Bag 380g"
##  [65] "Doritos Corn Chips  Nacho Cheese 170g"
##  [66] "Kettle Sensations   BBQ&Maple 150g"
##  [67] "WW D/Style Chip     Sea Salt 200g"
##  [68] "Pringles Chicken    Salt Crips 134g"
##  [69] "WW Original Stacked Chips 160g"
##  [70] "Smiths Chip Thinly  CutSalt/Vinegr175g"
##  [71] "Cheezels Cheese 330g"
##  [72] "Tostitos Lightly    Salted 175g"
##  [73] "Thins Chips Salt &  Vinegar 175g"
##  [74] "Smiths Crinkle Cut  Chips Barbecue 170g"
##  [75] "Cheetos Puffs 165g"
##  [76] "RRD Sweet Chilli &  Sour Cream 165g"
##  [77] "WW Crinkle Cut      Original 175g"
##  [78] "Tostitos Splash Of  Lime 175g"
##  [79] "Kettle Tortilla ChpsBtroot&Ricotta 150g"
##  [80] "CCs Tasty Cheese    175g"
##  [81] "Woolworths Cheese   Rings 190g"
##  [82] "Tostitos Smoked     Chipotle 175g"
##  [83] "Pringles Barbeque   134g"
##  [84] "WW Supreme Cheese   Corn Chips 200g"
##  [85] "Pringles Mystery    Flavour 134g"
##  [86] "Tyrrells Crisps     Ched & Chives 165g"
##  [87] "Snbts Whlgrn Crisps Cheddr&Mstrd 90g"
##  [88] "Cheetos Chs & Bacon Balls 190g"
##  [89] "Pringles Slt Vingar 134g"
##  [90] "Infuzions SourCream&Herbs Veg Strws 110g"
##  [91] "Kettle Tortilla ChpsFeta&Garlic 150g"
##  [92] "Infuzions Mango     Chutny Papadums 70g"
##  [93] "RRD Steak &         Chimuchurri 150g"
##  [94] "RRD Honey Soy       Chicken 165g"
##  [95] "Sunbites Whlegrn    Crisps Frch/Onin 90g"
##  [96] "RRD Salt & Vinegar  165g"
##  [97] "Doritos Cheese      Supreme 330g"
##  [98] "Smiths Crinkle Cut  Snag&Sauce 150g"
##  [99] "WW Sour Cream &OnionStacked Chips 160g"
## [100] "RRD Lime & Pepper   165g"
## [101] "Natural ChipCo Sea  Salt & Vinegr 175g"
## [102] "Red Rock Deli Chikn&Garlic Aioli 150g"
## [103] "RRD SR Slow Rst     Pork Belly 150g"
## [104] "RRD Pc Sea Salt     165g"
## [105] "Smith Crinkle Cut   Bolognese 150g"
```

```
# Summarize the data to check for nulls and possible 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
```

```
# Filter the dataset to find the outlier where 200 packets of chips are bought
outlier_transactions <- transactionData[PROD_QTY == 200]
print(outlier_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
```

```
# Find the customer who bought 200 packets of chips
outlier_customer <- outlier_transactions$LYLTY_CARD_NBR

# Use a filter to see what other transactions that customer made
customer_transactions <- transactionData[LYLTY_CARD_NBR %in% outlier_customer]
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
```

```
# Remove the outliers
transactionData <- transactionData[PROD_QTY != 200]

# Verify the removal
summary(transactionData$PROD_QTY)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   2.000   2.000   1.906   2.000   5.000
```

```r
# Summarize the data to check for nulls and possible 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 : 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
```

```r
# Count the number of transactions by date
transaction_counts_by_date <- transactionData[, .N, by = DATE]

# Print the summary of transaction counts by date
print(transaction_counts_by_date)
```

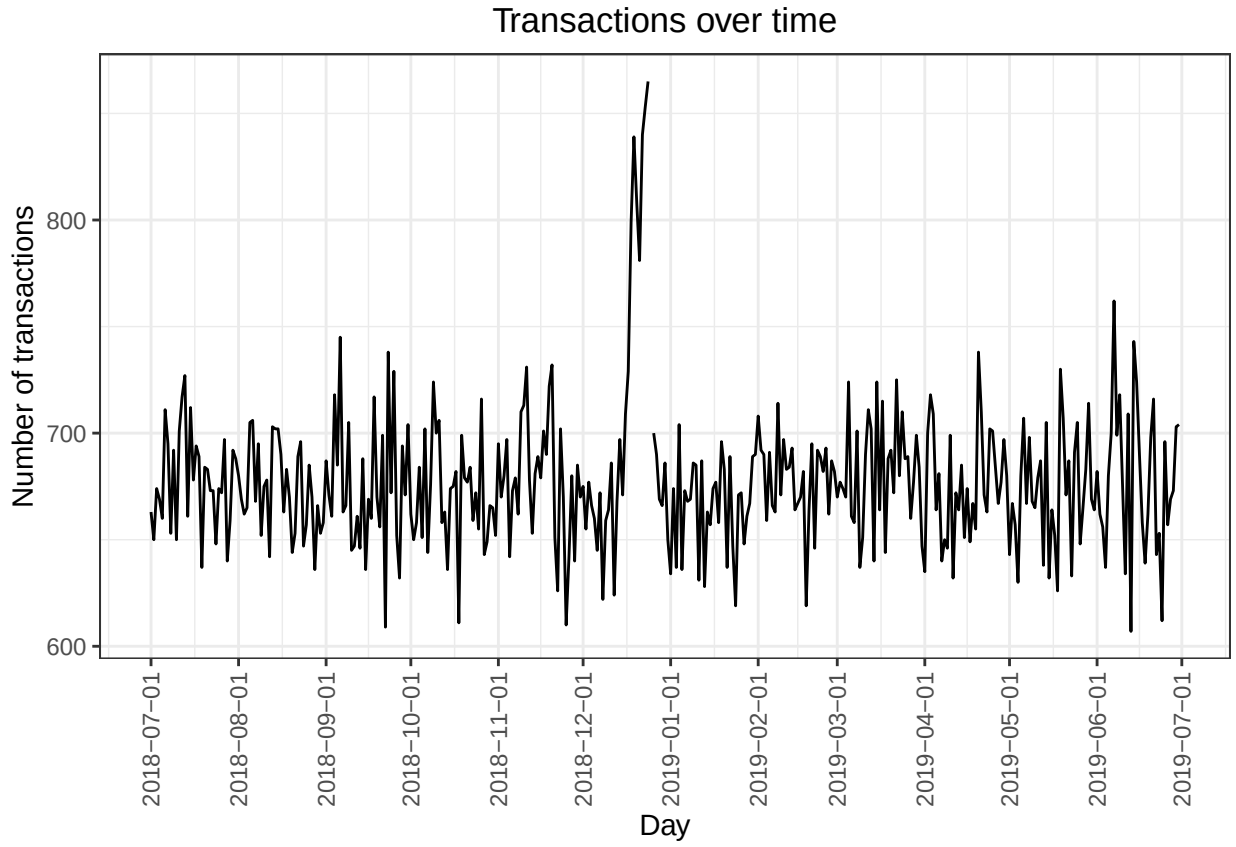
```
##            DATE     N
##          <Date> <int>
##   1: 2018-10-17   682
##   2: 2019-05-14   705
##   3: 2019-05-20   707
##   4: 2018-08-17   663
##   5: 2018-08-18   683
##  ---
## 360: 2018-12-08   622
## 361: 2019-01-30   689
## 362: 2019-02-09   671
## 363: 2018-08-31   658
## 364: 2019-02-12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

```r
# Optionally, summarize the transaction counts to check for irregularities
summary(transaction_counts_by_date$N)
```

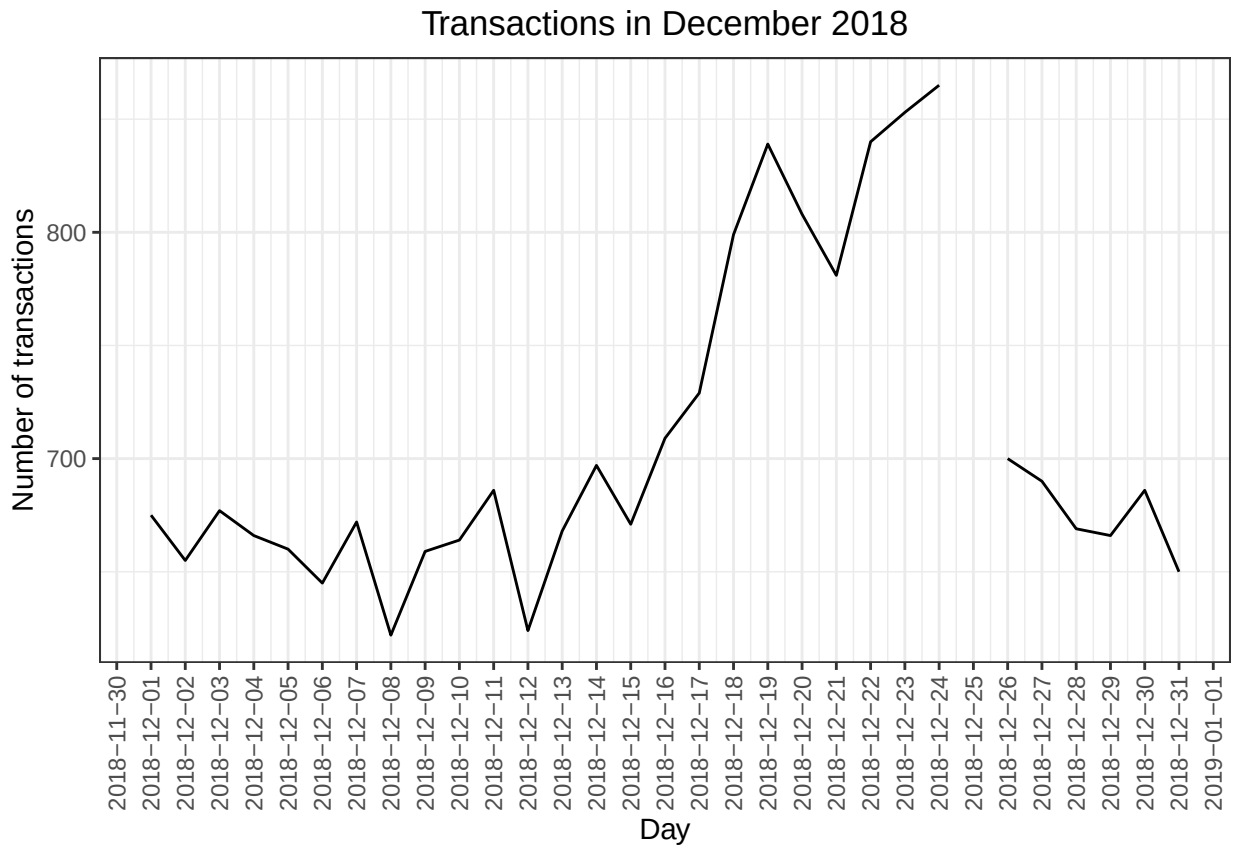
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   607.0   658.0   674.0   677.9   694.2   865.0
```

```r
# Create a sequence of dates from 1 Jul 2018 to 30 Jun 2019
date_sequence <- data.table(DATE = seq(as.Date("2018-07-01"),
 ↪  as.Date("2019-06-30"), by = "day"))

# Join this sequence with the transaction count data
transactions_by_day <- merge(date_sequence, transaction_counts_by_date, by =
 ↪  "DATE", all.x = TRUE)
```

```r
# Plot the number of transactions over time
theme_set(theme_bw())
theme_update(plot.title = element_text(hjust = 0.5))

ggplot(transactions_by_day, 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))
```




```r
# Filter the data to include only transactions in December 2018
december_transactions <- transactions_by_day[DATE >= as.Date("2018-12-01") &
 ↪  DATE <= as.Date("2018-12-31")]

# Plot transactions over time for December 2018
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))
```

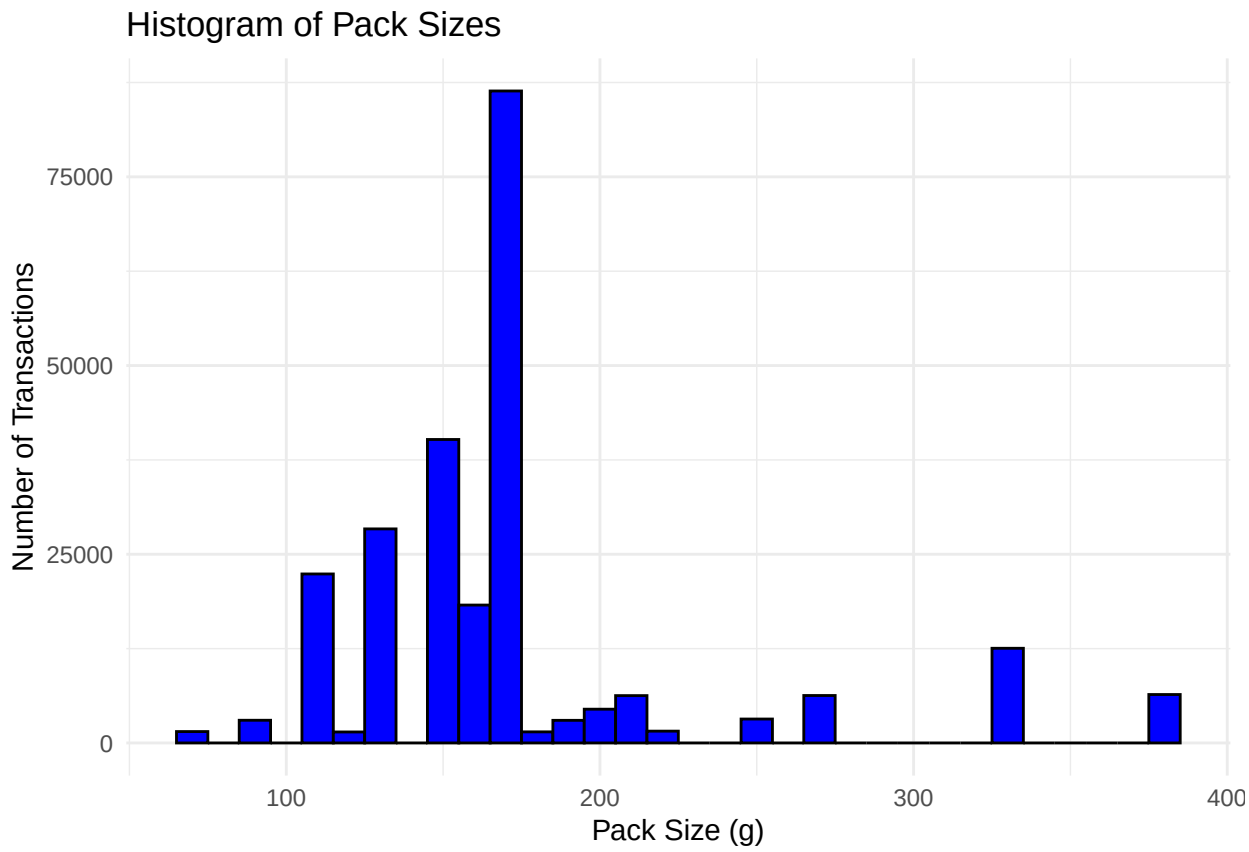



```
# Extract Pack Size
transactionData[, PACK_SIZE := parse_number(PROD_NAME)]

# Verify the Pack Sizes
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 of Pack Sizes

ggplot(transactionData, aes(x = PACK_SIZE)) +
  geom_histogram(binwidth = 10, fill = "blue", color = "black") +
  labs(title = "Histogram of Pack Sizes", x = "Pack Size (g)", y = "Number of
   ↪  Transactions") +
  theme_minimal()
```




```
# Extract the brand name from the PROD_NAME column
transactionData[, BRAND := toupper(word(PROD_NAME, 1))]

#Verify the brand names
transactionData[, .N, BRAND][order(-N)]
```

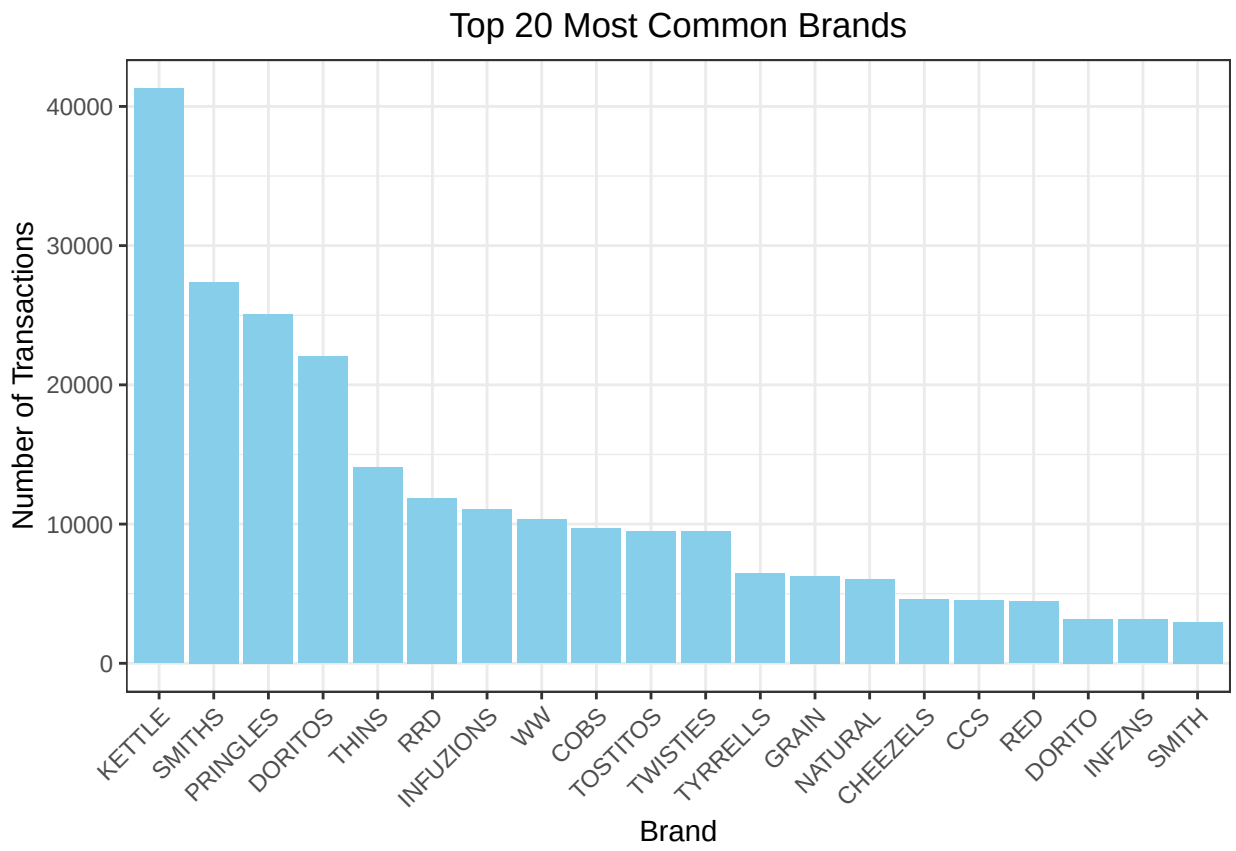
```
##          BRAND     N
##         <char> <int>
##  1:     KETTLE 41288
##  2:     SMITHS 27390
##  3:   PRINGLES 25102
```

```
##  4:    DORITOS 22041
##  5:      THINS 14075
##  6:        RRD 11894
##  7:  INFUZIONS 11057
##  8:         WW 10320
##  9:       COBS  9693
## 10:   TOSTITOS  9471
## 11:   TWISTIES  9454
## 12:   TYRRELLS  6442
## 13:      GRAIN  6272
## 14:    NATURAL  6050
## 15:   CHEEZELS  4603
## 16:        CCS  4551
## 17:        RED  4427
## 18:     DORITO  3183
## 19:     INFZNS  3144
## 20:      SMITH  2963
## 21:    CHEETOS  2927
## 22:      SNBTS  1576
## 23:     BURGER  1564
## 24: WOOLWORTHS  1516
## 25:    GRNWVES  1468
## 26:   SUNBITES  1432
## 27:        NCC  1419
## 28:     FRENCH  1418
##          BRAND     N
```

```
# Check the number of unique brands and their frequency
brand_summary <- transactionData[, .N, BRAND][order(-N)]

#Check if there are any anomalies
anomalies <- transactionData[, .N, BRAND][N < 10]  # Assuming anomalies are
 ↪  brands with less than 10 transactions

print(anomalies)
```

```
## Empty data.table (0 rows and 2 cols): BRAND,N
```

```
# Plot the distribution of the most common brands

ggplot(head(brand_summary, 20), aes(x = reorder(BRAND, -N), y = N)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(title = "Top 20 Most Common Brands", x = "Brand", y = "Number of
   ↪  Transactions") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Clean brand names
transactionData[BRAND == "RED", BRAND := "RRD"]

# Add any additional brand adjustments
transactionData[BRAND == "SMITH", BRAND := "SMITHS"]  # Assuming "SMITH" and
 ↪  "SMITHS" are the same
transactionData[BRAND == "NCC", BRAND := "NATURAL"]   # Example if "NCC" and
 ↪  "NATURAL" refer to the same brand
transactionData[BRAND == "SNBTS", BRAND := "SUNBITES"]  # Merging possible
 ↪  variations

# Check the results again
brand_summary_cleaned <- transactionData[, .N, BRAND][order(-N)]

# View the top 10 most common brands after cleaning
print(head(brand_summary_cleaned, 10))
```

```
##         BRAND     N
##        <char> <int>
##  1:    KETTLE 41288
##  2:    SMITHS 30353
##  3:  PRINGLES 25102
##  4:   DORITOS 22041
##  5:       RRD 16321
##  6:     THINS 14075
```

```
##  7: INFUZIONS 11057
##  8:        WW 10320
##  9:      COBS  9693
## 10:  TOSTITOS  9471
```

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

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" "OLD
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
head(customerData, n=10)
```

```
##     LYLTY_CARD_NBR              LIFESTAGE PREMIUM_CUSTOMER
##              <int>                 <char>           <char>
##  1:           1000  YOUNG SINGLES/COUPLES          Premium
##  2:           1002  YOUNG SINGLES/COUPLES       Mainstream
##  3:           1003         YOUNG FAMILIES           Budget
##  4:           1004  OLDER SINGLES/COUPLES       Mainstream
##  5:           1005 MIDAGE SINGLES/COUPLES       Mainstream
##  6:           1007  YOUNG SINGLES/COUPLES           Budget
##  7:           1009           NEW FAMILIES          Premium
##  8:           1010  YOUNG SINGLES/COUPLES       Mainstream
##  9:           1011  OLDER SINGLES/COUPLES       Mainstream
## 10:           1012         OLDER FAMILIES       Mainstream
```

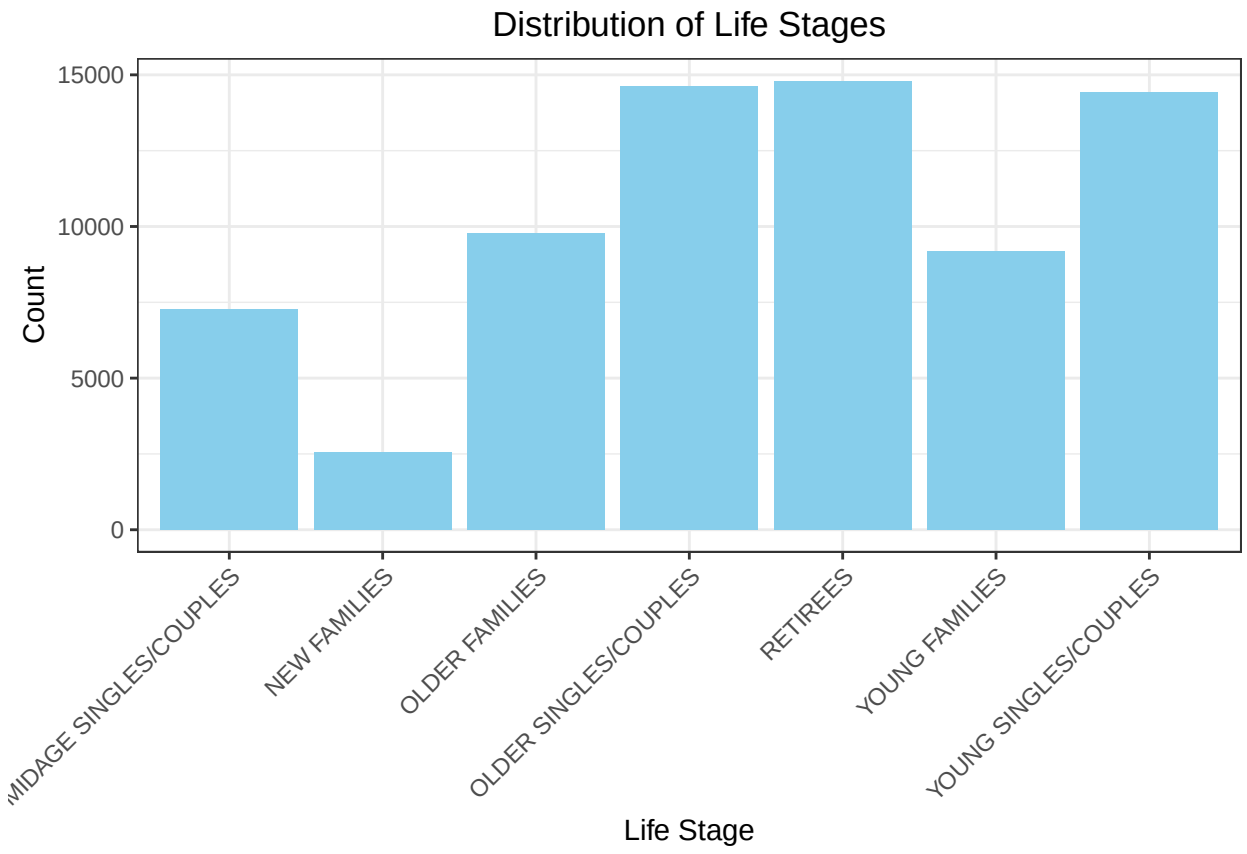
```
# Summary of the entire dataset
summary(customerData)
```

```
##  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
```

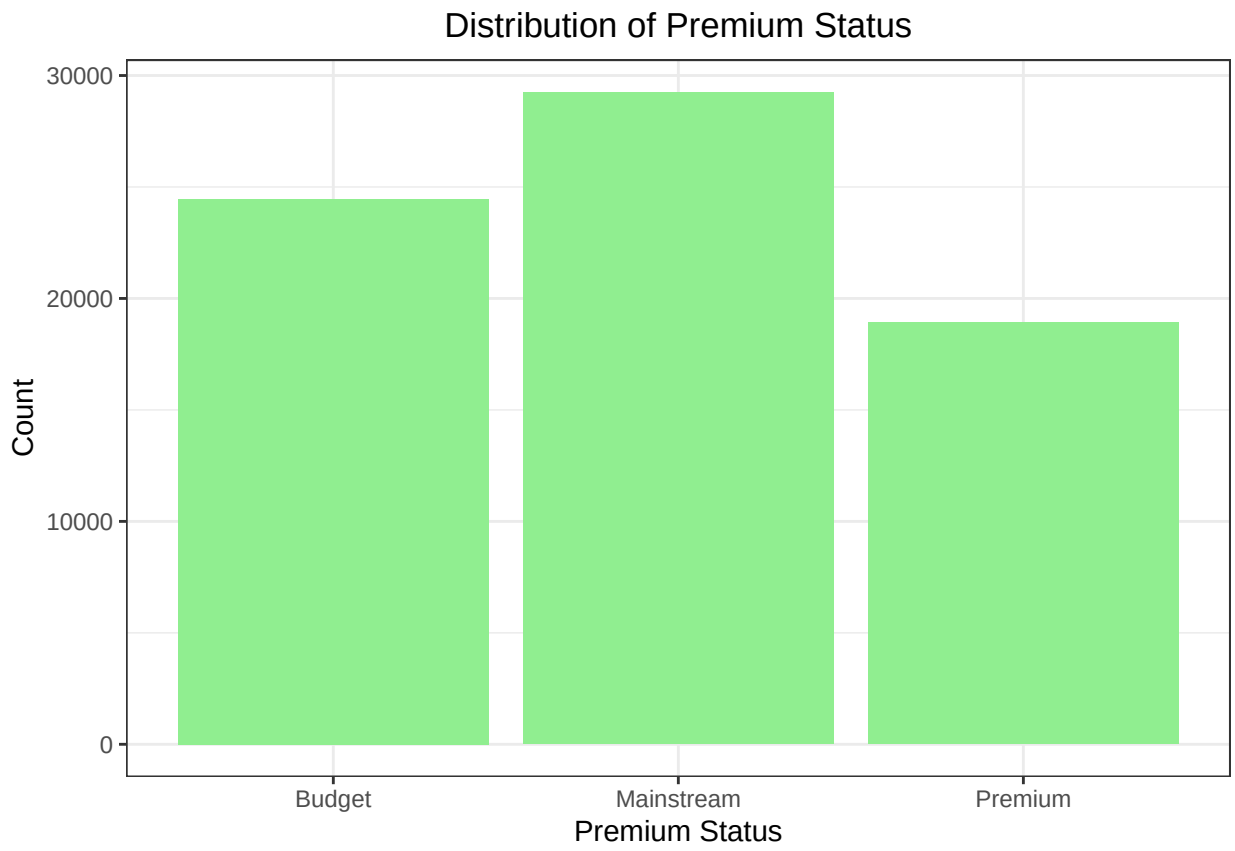
```
# Count the number of unique customers
num_customers <- customerData[, uniqueN(LYLTY_CARD_NBR)]
print(paste("Number of unique customers:", num_customers))
```

```
## [1] "Number of unique customers: 72637"
```

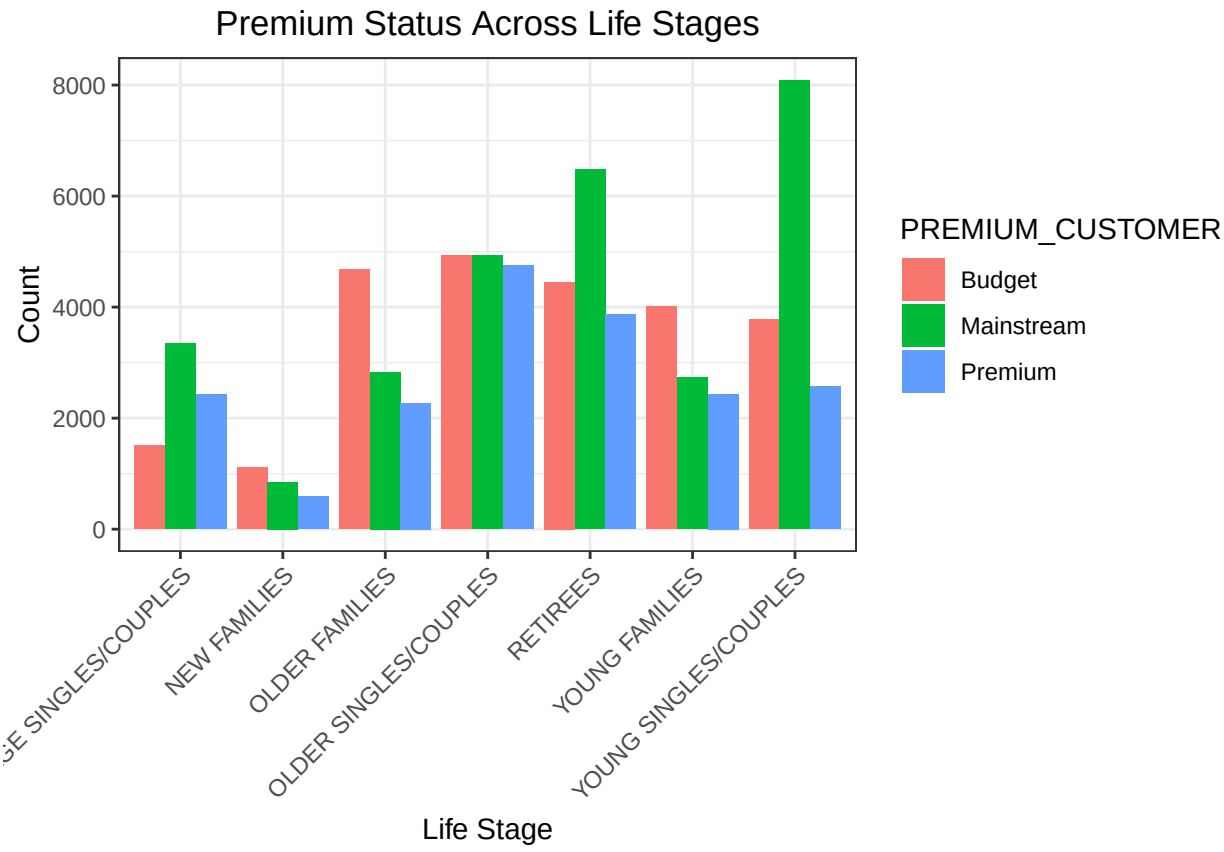
```r
# Distribution of key columns

# Distribution of Life Stages
ggplot(customerData, aes(x = LIFESTAGE)) +
  geom_bar(fill = "skyblue") +
  labs(title = "Distribution of Life Stages", x = "Life Stage", y = "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```r
# Distribution of Premium Status
ggplot(customerData, aes(x = PREMIUM_CUSTOMER)) +
  geom_bar(fill = "lightgreen") +
  labs(title = "Distribution of Premium Status", x = "Premium Status", y =
    ↪  "Count")
```




```r
# Cross-tabulation of Life Stage and Premium Status
ggplot(customerData, aes(x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) +
  geom_bar(position = "dodge") +
  labs(title = "Premium Status Across Life Stages", x = "Life Stage", y =
   ↪  "Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




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

# Checking for missing customer details after the merge
missing_customers <- data[is.na(LIFESTAGE) | is.na(PREMIUM_CUSTOMER)]

# Count the number of transactions without matched customer details
num_missing <- nrow(missing_customers)
print(paste("Number of transactions with missing customer details:",
 ↪  num_missing))
```

```
## [1] "Number of transactions with missing customer details: 0"
```

```
#Save the merged dataset as CSV for Task 2
fwrite(data, paste0(filePath,"QVI_data.csv"))

Merged <- fread(paste0(filePath,"QVI_data.csv"))
str(Merged)
```

```
## Classes 'data.table' and 'data.frame':   246740 obs. of  12 variables:
##  $ LYLTY_CARD_NBR  : int  1000 1002 1003 1003 1004 1005 1007 1007 1009 1010 ...
##  $ DATE            : IDate, format: "2018-10-17" "2018-09-16" ...
##  $ STORE_NBR       : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TXN_ID          : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
##  $ PROD_NBR        : int  5 58 52 106 96 86 49 10 20 51 ...
## $ PROD_NAME     : chr  "Natural Chip      Compny SeaSalt175g" "Red Rock Deli Chikn&Garlic Aio
##  $ PROD_QTY        : int  2 1 1 1 1 1 1 1 1 2 ...
##  $ TOT_SALES       : num  6 2.7 3.6 3 1.9 2.8 3.8 2.7 5.7 8.8 ...
##  $ PACK_SIZE       : int  175 150 210 175 160 165 110 150 330 170 ...
##  $ BRAND           : chr  "NATURAL" "RRD" "GRAIN" "NATURAL" ...
## $ LIFESTAGE     : chr  "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "YOU
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Budget" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
head(Merged)
```

```
##    LYLTY_CARD_NBR       DATE STORE_NBR TXN_ID PROD_NBR
##             <int>     <IDat>     <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   BRAND
##                                  <char>    <int>     <num>     <int>  <char>
## 1: Natural Chip        Compny SeaSalt175g        2       6.0       175 NATURAL
## 2:  Red Rock Deli Chikn&Garlic Aioli 150g        1       2.7       150     RRD
## 3:  Grain Waves Sour    Cream&Chives 210G        1       3.6       210   GRAIN
## 4: Natural ChipCo      Hony Soy Chckn175g        1       3.0       175 NATURAL
## 5:         WW Original Stacked Chips 160g        1       1.9       160      WW
## 6:                     Cheetos Puffs 165g        1       2.8       165 CHEETOS
##                 LIFESTAGE PREMIUM_CUSTOMER
##                     <char>           <char>
## 1:  YOUNG SINGLES/COUPLES          Premium
## 2:  YOUNG SINGLES/COUPLES       Mainstream
## 3:         YOUNG FAMILIES           Budget
## 4:         YOUNG FAMILIES           Budget
## 5:  OLDER SINGLES/COUPLES       Mainstream
## 6: MIDAGE SINGLES/COUPLES       Mainstream
```

```
summary(Merged)
```

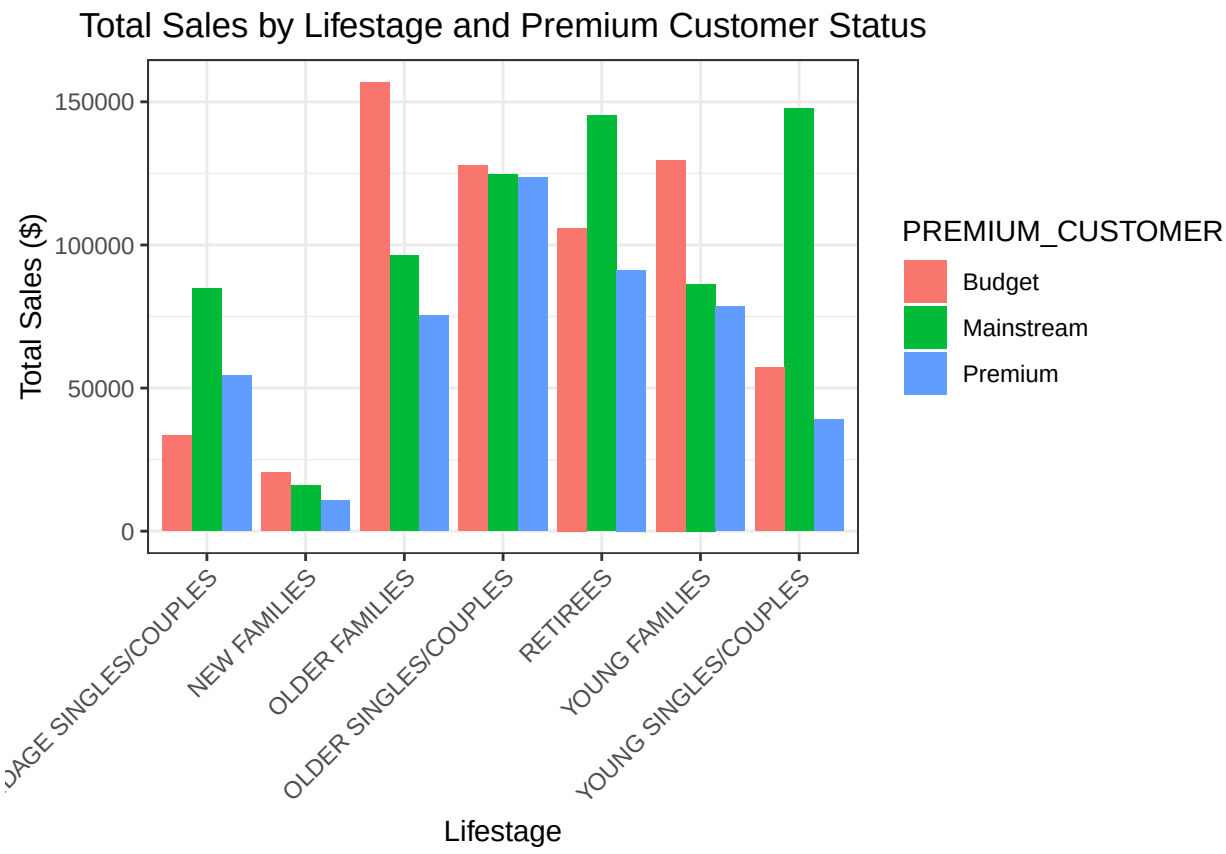
```
##  LYLTY_CARD_NBR         DATE               STORE_NBR         TXN_ID
##  Min.   :   1000   Min.   :2018-07-01   Min.   :  1.0   Min.   :      1
##  1st Qu.:  70015   1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  67569
##  Median : 130367   Median :2018-12-30   Median :130.0   Median : 135182
##  Mean   : 135530   Mean   :2018-12-30   Mean   :135.1   Mean   : 135130
##  3rd Qu.: 203083   3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 202652
##  Max.   :2373711   Max.   :2019-06-30   Max.   :272.0   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
##    PACK_SIZE        BRAND            LIFESTAGE         PREMIUM_CUSTOMER
##  Min.   : 70.0   Length:246740      Length:246740      Length:246740
##  1st Qu.:150.0   Class :character   Class :character   Class :character
##  Median :170.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :175.6
##  3rd Qu.:175.0
##  Max.   :380.0
```

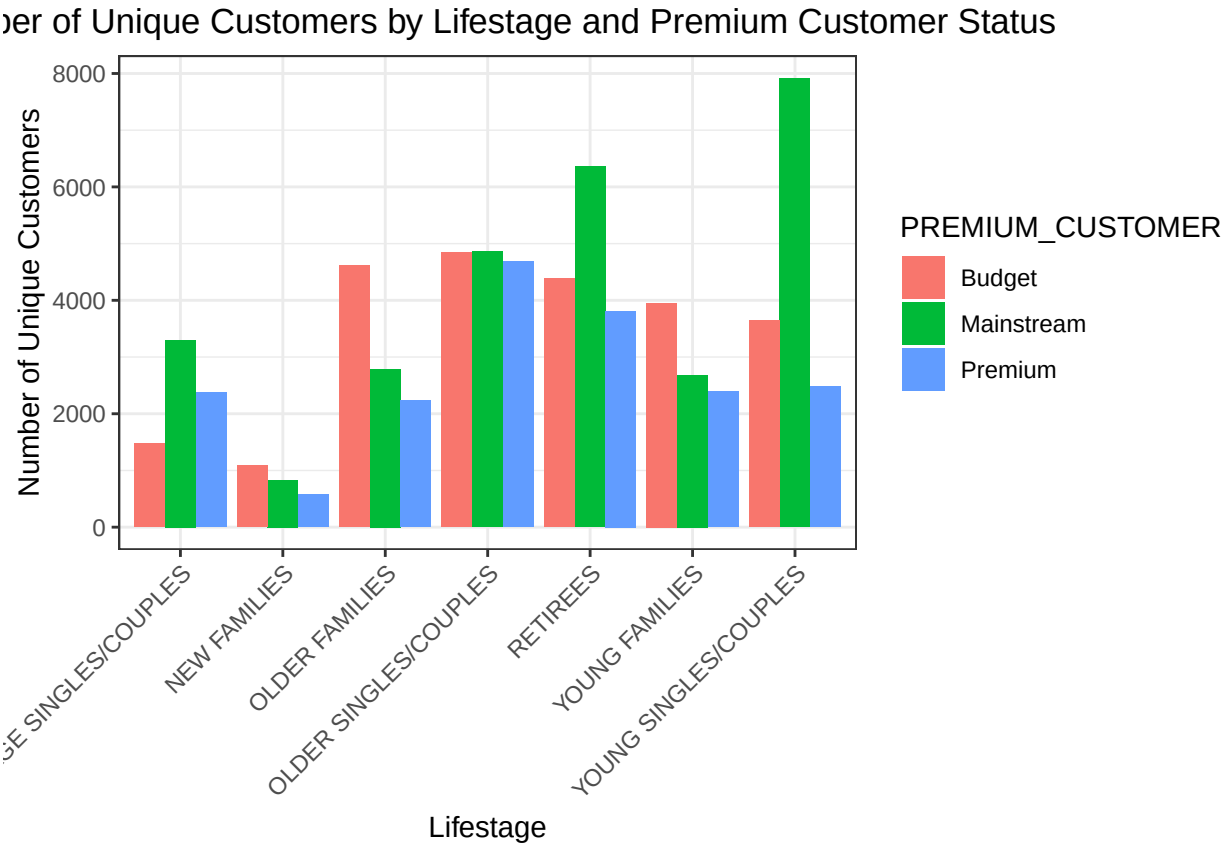
```
# Calculate Total Sales by LIFESTAGE and PREMIUM_CUSTOMER
total_sales_by_segment <- data[, .(Total_Sales = sum(TOT_SALES)), by =
 ↪  .(LIFESTAGE, PREMIUM_CUSTOMER)]

# View the summary
print(total_sales_by_segment)
```

```
##                  LIFESTAGE PREMIUM_CUSTOMER Total_Sales
##                     <char>           <char>       <num>
##  1:  YOUNG SINGLES/COUPLES          Premium    39052.30
##  2:  YOUNG SINGLES/COUPLES       Mainstream   147582.20
##  3:          YOUNG FAMILIES           Budget   129717.95
##  4:  OLDER SINGLES/COUPLES       Mainstream   124648.50
##  5: MIDAGE SINGLES/COUPLES       Mainstream    84734.25
##  6:  YOUNG SINGLES/COUPLES           Budget    57122.10
##  7:            NEW FAMILIES          Premium    10760.80
##  8:          OLDER FAMILIES       Mainstream    96413.55
##  9:                 RETIREES           Budget   105916.30
## 10:  OLDER SINGLES/COUPLES          Premium   123537.55
## 11:          OLDER FAMILIES           Budget   156863.75
## 12: MIDAGE SINGLES/COUPLES          Premium    54443.85
## 13:          OLDER FAMILIES          Premium    75242.60
## 14:                 RETIREES       Mainstream   145168.95
## 15:                 RETIREES          Premium    91296.65
## 16:          YOUNG FAMILIES       Mainstream    86338.25
## 17: MIDAGE SINGLES/COUPLES           Budget    33345.70
## 18:            NEW FAMILIES       Mainstream    15979.70
## 19:  OLDER SINGLES/COUPLES           Budget   127833.60
## 20:          YOUNG FAMILIES          Premium    78571.70
## 21:            NEW FAMILIES           Budget    20607.45
##                  LIFESTAGE PREMIUM_CUSTOMER Total_Sales
```

```
# Plot the results
ggplot(total_sales_by_segment, aes(x = LIFESTAGE, y = Total_Sales, fill =
 ↪  PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Total Sales by Lifestage and Premium Customer Status",
       x = "Lifestage",
       y = "Total Sales ($)") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

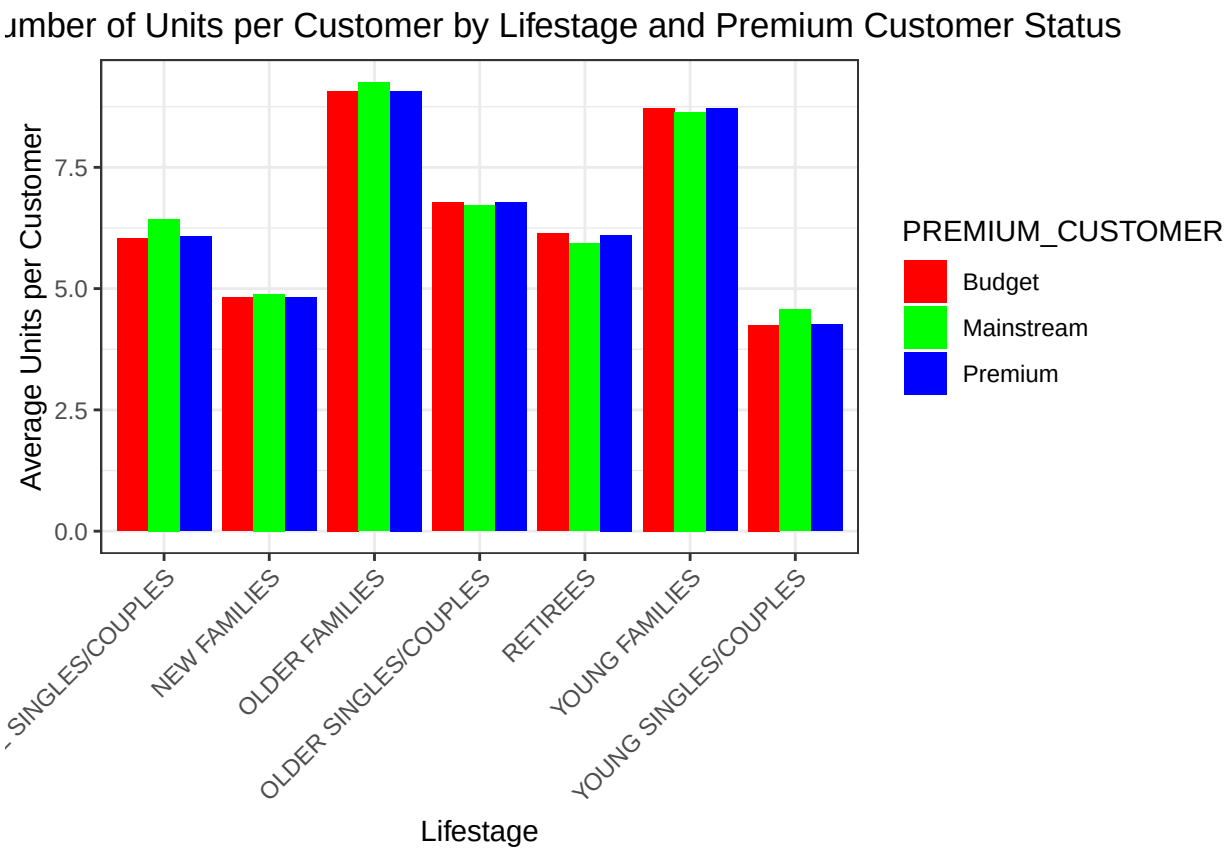



```
# Number of customers by LIFESTAGE and PREMIUM_CUSTOMER
customers_by_segment <- data[, .(Unique_Customers = uniqueN(LYLTY_CARD_NBR)),
 ↪  by = .(LIFESTAGE, PREMIUM_CUSTOMER)]

# Plot the results
ggplot(customers_by_segment, aes(x = LIFESTAGE, y = Unique_Customers, fill =
 ↪  PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Number of Unique Customers by Lifestage and Premium Customer
   ↪  Status",
       x = "Lifestage",
       y = "Number of Unique Customers") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




```
# Calculate total units sold by LIFESTAGE and PREMIUM_CUSTOMER
total_units <- data[, .(total_units = sum(PROD_QTY)), by = .(LIFESTAGE,
 ↪  PREMIUM_CUSTOMER)]

# Calculate the number of unique customers in each LIFESTAGE and
 ↪  PREMIUM_CUSTOMER segment
unique_customers <- data[, .(unique_customers = uniqueN(LYLTY_CARD_NBR)), by =
 ↪  .(LIFESTAGE, PREMIUM_CUSTOMER)]

# Merge the total_units and unique_customers data
units_per_customer <- merge(total_units, unique_customers, by = c("LIFESTAGE",
 ↪  "PREMIUM_CUSTOMER"))

# Calculate the average units per customer
units_per_customer[, avg_units_per_customer := total_units / unique_customers]
```

```
# Plot the average number of units per customer
ggplot(units_per_customer, aes(x = LIFESTAGE, y = avg_units_per_customer, fill
 ↪  = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Number of Units per Customer by Lifestage and Premium
   ↪  Customer Status",
       x = "Lifestage", y = "Average Units per Customer") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("red", "green", "blue"))
```




```r
# Calculate average price per unit using aggregate (base R)
avg_price_per_unit <- aggregate(TOT_SALES / PROD_QTY ~ LIFESTAGE +
 ↪  PREMIUM_CUSTOMER,
                                data = data,
                                FUN = mean,
                                na.rm = TRUE)

# Rename the calculated column for clarity
colnames(avg_price_per_unit)[3] <- "Average_Price_Per_Unit"

# Plot the results
ggplot(avg_price_per_unit, aes(x = LIFESTAGE, y = Average_Price_Per_Unit, fill
 ↪  = PREMIUM_CUSTOMER)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(title = "Average Price per Unit by Lifestage and Premium Customer
   ↪  Status",
       x = "Lifestage",
       y = "Average Price per Unit") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```


```
# Filter the data for 'Midage Singles/Couples' and 'Young Singles/Couples'
filtered_data <- subset(data, LIFESTAGE %in% c("MIDAGE SINGLES/COUPLES", "YOUNG
 ↪  SINGLES/COUPLES"))

# Split the data by PREMIUM_CUSTOMER type (Mainstream vs Premium & Budget)
mainstream_data <- subset(filtered_data, PREMIUM_CUSTOMER == "Mainstream")
premium_data <- subset(filtered_data, PREMIUM_CUSTOMER == "Premium")
budget_data <- subset(filtered_data, PREMIUM_CUSTOMER == "Budget")

# Perform a t-test between Mainstream vs Premium
t_test_mainstream_vs_premium <- t.test(mainstream_data$TOT_SALES /
 ↪  mainstream_data$PROD_QTY,
                                       premium_data$TOT_SALES /
 ↪  premium_data$PROD_QTY)
```

```
# Perform a t-test between Mainstream vs Budget
t_test_mainstream_vs_budget <- t.test(mainstream_data$TOT_SALES /
 ↪  mainstream_data$PROD_QTY,
                                     budget_data$TOT_SALES / budget_data$PROD_QTY)

# Print the t-test results
t_test_mainstream_vs_premium
```

```
##
##  Welch Two Sample t-test
```

```
##
## data:  mainstream_data$TOT_SALES/mainstream_data$PROD_QTY and premium_data$TOT_SALES/premi
## t = 28.338, df = 23872, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.2930718 0.3366263
## sample estimates:
## mean of x mean of y
##  4.039786  3.724937
```

```
t_test_mainstream_vs_budget
```

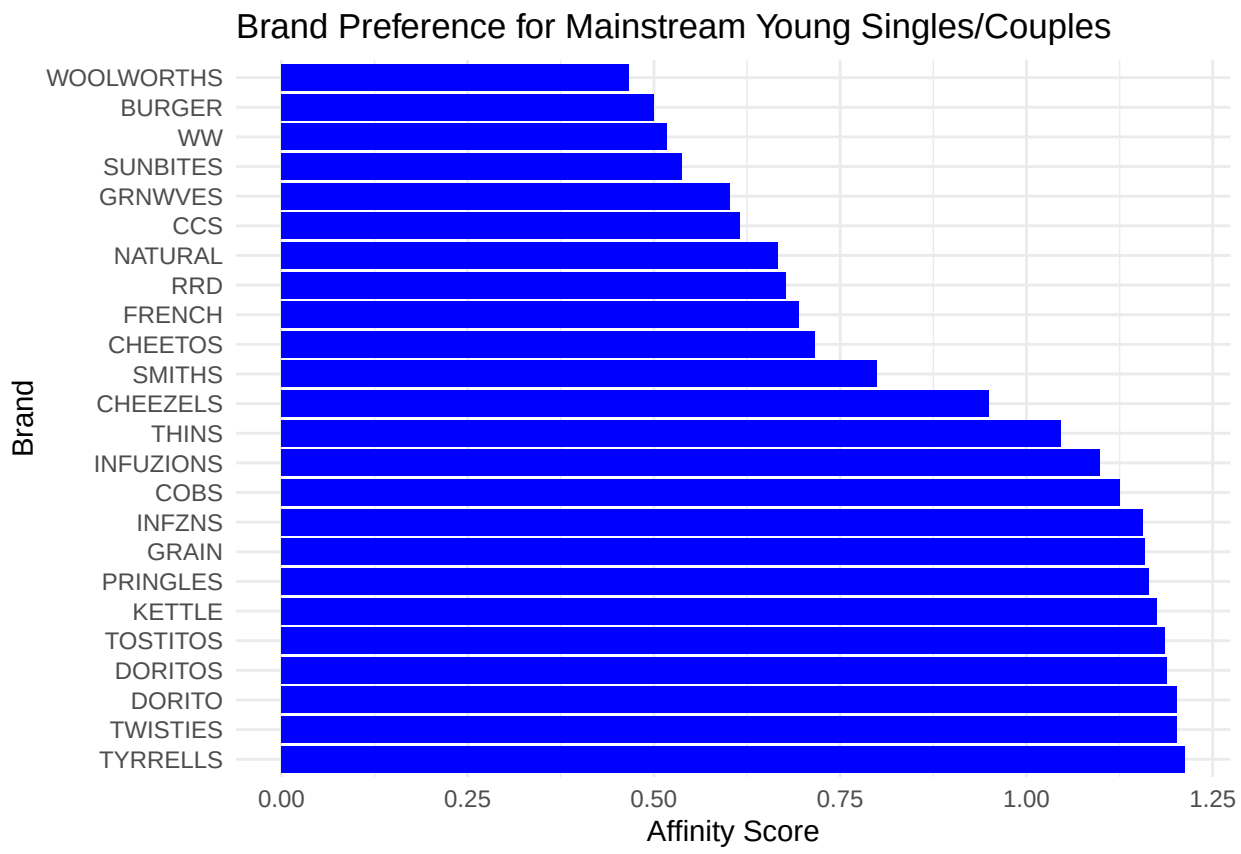
```
##
##  Welch Two Sample t-test
##
## data:  mainstream_data$TOT_SALES/mainstream_data$PROD_QTY and budget_data$TOT_SALES/budget_
## t = 31.671, df = 23526, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.3302324 0.3738041
## sample estimates:
## mean of x mean of y
##  4.039786  3.687768
```

```
# Filter data for Mainstream - young singles/couples
mainstream_young <- subset(data, LIFESTAGE == "YOUNG SINGLES/COUPLES" &
 ↪  PREMIUM_CUSTOMER == "Mainstream")

#Calculate brand preference within this segment
mainstream_young_brand_freq <- table(mainstream_young$BRAND) /
 ↪  nrow(mainstream_young)

# Calculate overall brand preference across all customer segments
overall_brand_freq <- table(data$BRAND) / nrow(data)
```

```
# Calculate the affinity score (brand preference ratio for the segment vs.
 ↪  overall)
affinity_score <- mainstream_young_brand_freq / overall_brand_freq

# Sort and display the top brands preferred by this segment
affinity_score <- sort(affinity_score, decreasing = TRUE)
affinity_score
```

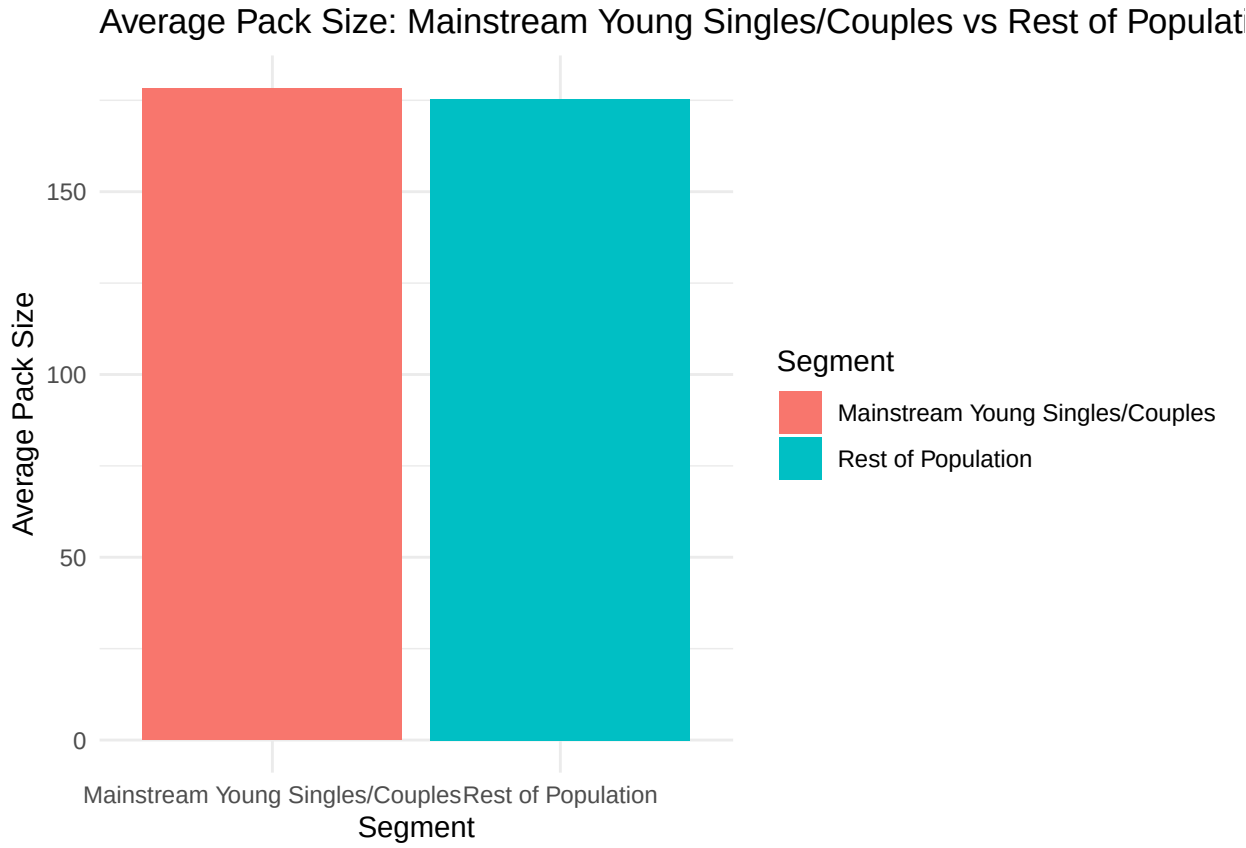
```
##
##   TYRRELLS   TWISTIES     DORITO    DORITOS   TOSTITOS     KETTLE   PRINGLES
##  1.2130984  1.2018576  1.2017997  1.1891104  1.1863703  1.1753999  1.1643104
##      GRAIN     INFZNS       COBS  INFUZIONS      THINS   CHEEZELS     SMITHS
##  1.1594247  1.1564745  1.1253345  1.0984085  1.0458665  0.9489891  0.7990093
##    CHEETOS     FRENCH        RRD    NATURAL        CCS    GRNWVES   SUNBITES
##  0.7159974  0.6944556  0.6768421  0.6659780  0.6158462  0.6020022  0.5372275
##         WW     BURGER WOOLWORTHS
##  0.5174719  0.5004735  0.4663532
```

```r
# Convert affinity_score into a data frame for easy plotting
affinity_df <- data.frame(Brand = names(affinity_score), Affinity =
 ↪  as.numeric(affinity_score))

# Plot the affinity scores
ggplot(affinity_df, aes(x = reorder(Brand, -Affinity), y = Affinity)) +
  geom_bar(stat = "identity", fill = "blue") +
  coord_flip() +
  labs(title = "Brand Preference for Mainstream Young Singles/Couples",
       x = "Brand",
       y = "Affinity Score") +
  theme_minimal()
```




```r
# Filter data for Mainstream Young Singles/Couples and the rest
mainstream_young <- subset(data, LIFESTAGE == "YOUNG SINGLES/COUPLES" &
 ↪  PREMIUM_CUSTOMER == "Mainstream")
rest_of_population <- subset(data, !(LIFESTAGE == "YOUNG SINGLES/COUPLES" &
 ↪  PREMIUM_CUSTOMER == "Mainstream"))

# Calculate average pack size for Mainstream Young Singles/Couples
avg_pack_size_mainstream_young <- mean(mainstream_young$PACK_SIZE)

# Calculate average pack size for the rest of the population
avg_pack_size_rest <- mean(rest_of_population$PACK_SIZE)
```

```r
# Plotting the comparison
pack_size_data <- data.frame(
  Segment = c("Mainstream Young Singles/Couples", "Rest of Population"),
  Avg_Pack_Size = c(avg_pack_size_mainstream_young, avg_pack_size_rest)
)


ggplot(pack_size_data, aes(x = Segment, y = Avg_Pack_Size, fill = Segment)) +
  geom_bar(stat = "identity") +
  ggtitle("Average Pack Size: Mainstream Young Singles/Couples vs Rest of
   ↪  Population") +
  xlab("Segment") +
  ylab("Average Pack Size") +
  theme_minimal()
```



Conclusion

In conclusion, **Mainstream Young Singles/Couples** prefer specific brands like **Tyrrells, Twisties, and Doritos** over others, indicating brand loyalty in this segment. They also tend to purchase slightly **larger pack sizes** compared to the rest of the population, likely due to their preference for bulk buying, possibly for social or entertainment purposes. These insights suggest targeting this group with brand-specific promotions and larger pack options could increase sales retention and growth within this segment.