Task 1: Customer Trends & Target Segment, Jul 2018 - Jun 2019

Ed Garcia

8/17/2021

At the request of the Chips Category Manager, I am exploring and analyzing 1 year of transaction and customer data related to chips sales in an Australian retail chain. I am looking out for purchasing trends, target demographics, and insights for commercial recommendation to help the client develop their marketing strategy for the following year.

set options for R markdown knitting

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
```

LOAD REQUIRED LIBRARIES

```
library(data.table)
library(ggplot2)
library(ggmosaic)
library(tidyverse)
library(ggpubr)
```

IMPORT SOURCE CSV FILES INTO R

```
transactionData <- read.csv("C:/Users/garci/OneDrive/Desktop/Data Analysis Education/Forage Virtual Int
customerData <- read.csv("C:/Users/garci/OneDrive/Desktop/Data Analysis Education/Forage Virtual Interns
```

EXPLORATORY DATA ANALYSIS

Now that the data files are loaded, inspect them to gain a broad understanding of their size, structure, and content.

This will inform the exploratory data analysis.

```
str(transactionData)
```

```
## '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 ...
```

```
head(transactionData)
```

```
##    DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 1 43390         1           1000      1        5
## 2 43599         1           1307    348       66
## 3 43605         1           1343    383       61
## 4 43329         2           2373    974       69
## 5 43330         2           2426   1038      108
## 6 43604         4           4074   2982       57
##                                PROD_NAME PROD_QTY TOT_SALES
## 1   Natural Chip        Compny SeaSalt175g        2       6.0
## 2                CCs Nacho Cheese    175g        3       6.3
## 3   Smiths Crinkle Cut  Chips Chicken 170g        2       2.9
## 4   Smiths Chip Thinly  S/Cream&Onion 175g        5      15.0
## 5 Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13.8
## 6 Old El Paso Salsa   Dip Tomato Mild 300g        1       5.1
```

From first glance, this is how I would summarize the transactionData data frame:

- The date column refers to the date of each chips transaction. Also, the date is in a strange format.
- Store number is a store ID.
- Loyalty Card Number refers to individual customers.
- Transaction IDs are unique ID numbers that refer to individual transactions.
- Product number is a unique code assigned to each Product name.
- Product Quantity refers to the amount of products purchased in each transaction.
- Total sales is the amount of money spent on each transaction.

```
str(customerData)
```

```
## '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 SI
##  $ PREMIUM_CUSTOMER: chr  "Premium" "Mainstream" "Budget" "Mainstream" ...
```

```
head(customerData)
```

```
##   LYLTY_CARD_NBR              LIFESTAGE PREMIUM_CUSTOMER
## 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
```

From first glance, this is how I would summarize the customerData data frame:

- Loyalty Card Number refers to individual customers. This will be used to relate the transaction and customer data frames to each other.
- Lifestage refers to the customer demographic according to general age and family size.
- Premium Customer refers to the affluence level of the customer in regards to their general purchasing behavior Budget - they buy the cheapest brands, Mainstream - they buy standard-priced brands, Premium - they buy expensive brands. This category was predetermined by the client, and it is not exclusive to chips.

I will clean and explore the transaction data first.

CLEAN TRANSACTION DATA

Convert DATE column to a date format

After further exploration, the dates are listed in an Excel Date serial number format. The dates must be converted using this code:

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

```
##         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
## 6 2019-05-19         4           4074   2982       57
##                                PROD_NAME PROD_QTY TOT_SALES
## 1   Natural Chip        Compny SeaSalt175g        2       6.0
## 2                 CCs Nacho Cheese    175g        3       6.3
## 3   Smiths Crinkle Cut  Chips Chicken 170g        2       2.9
## 4   Smiths Chip Thinly  S/Cream&Onion 175g        5      15.0
## 5 Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13.8
## 6 Old El Paso Salsa   Dip Tomato Mild 300g        1       5.1
```

The dates are now in a readable format.

Examine PROD_NAME

How many different distinct product names are there?

```
n_distinct(transactionData$PROD_NAME)
```

```
## [1] 114
```

There are 114 distinct product names. Is this correct? I noticed in the head code I ran earlier that there is an product named "Old El Paso Salsa Dip Tomato Mild 300g". This does not sound like a chip name. I want to determine what keywords appear the most in the PROD_NAMES column. To do that, first I must split the strings in this column.

This splits the PROD_NAME strings into a separate data frame, using space as a delimiter:

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

This provides a count of each word:

```
text_wordcounts <- productWords %>%
  count(productWords$V1, sort = TRUE)
text_wordcounts
```

```
##      productWords$V1   n
##   1:                 234
##   2:            175g  26
##   3:           Chips  21
##   4:            150g  19
##   5:               &  17
##  ---
```

```
## 217:             Veg   1
## 218:          Vinegr   1
## 219:          Vingar   1
## 220:         Whlegrn   1
## 221:          Whlgrn   1
```

The word "Salsa" appears 9 times, "Dip" appears 3 times, and "OnionDip" appears 1 time. Is it possible that these are not chips categories?

Further explore the unique product names armed with this information:

```
uniqueProducts <- data.frame(unique(transactionData$PROD_NAME))
```

Use the filter in the View pane of UniqueProducts dataframe to search for the words "Salsa" and "Dip".

In the View pane of RStudio, filter and examine "Salsa":

After a Google search, I discovered that some of the products that contain the word "Salsa" are chips, and some are not chips:

Old El Paso Salsa Dip Tomato Mild = not chips Red Rock Deli SR Salsa & Mzzrlla 150g = chips Smiths Crinkle Cut Tomato Salsa 150g = chips Doritos Salsa Medium 300g = not chips Old El Paso Salsa Dip Chnky Tom Ht300g = not chips Woolworths Mild Salsa 300g = not chips Old El Paso Salsa Dip Tomato Med 300g = not chips Woolworths Medium Salsa 300g = not chips Doritos Salsa Mild 300g = not chips

I CANNOT REMOVE ALL PRODUCTS WITH THE WORD "SALSA" OR I WILL BE REMOVING SEVERAL TRANSACTIONS FROM THE DATASET THAT ARE CHIPS PRODUCTS.

The Chips Category Manager will not be happy if I provide insights on inaccurate data.

Filter (using View pane) and examine "Dip":

After a Google search, I discovered that all of the products that contain the word "Dip" are not chips.

Most of them were already included in the "Salsa" examination (and were found to be not chips).

There is one non-salsa dip:

Smiths Crinkle Cut French OnionDip 150g = not chips

I can safely remove all products with the word "Dip". These are not chips products.

Remove irrelevant data

Remove all dip products:

```
transactionData <- transactionData[!grepl("Dip", transactionData$PROD_NAME),]
```

Remove the actual salsa products:

```
transactionData <-
  transactionData[!grepl("Doritos Salsa       Medium 300g", transactionData$PROD_NAME),]
transactionData <-
  transactionData[!grepl("Woolworths Mild     Salsa 300g", transactionData$PROD_NAME),]
transactionData <-
  transactionData[!grepl("Woolworths Medium   Salsa 300g", transactionData$PROD_NAME),]
transactionData <-
  transactionData[!grepl("Doritos Salsa Mild  300g", transactionData$PROD_NAME),]
```

Re-examine PROD_NAME:

```
n_distinct(transactionData$PROD_NAME)
```

```
## [1] 106
```

There are now only 106 distinct product names. The 8 salsa/dip products have been removed.

REMOVE NULLS AND OUTLIERS FROM TRANSACTION DATA

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   Min.   :   1000   Min.   :      1
##  1st Qu.:2018-09-30   1st Qu.: 70   1st Qu.:  70015   1st Qu.:  67565
##  Median :2018-12-30   Median :130   Median : 130360   Median : 135151
##  Mean   :2018-12-30   Mean   :135   Mean   : 135524   Mean   : 135126
##  3rd Qu.:2019-03-31   3rd Qu.:203   3rd Qu.: 203082   3rd Qu.: 202645
##  Max.   :2019-06-30   Max.   :272   Max.   :2373711   Max.   :2415841
##     PROD_NBR    PROD_NAME            PROD_QTY         TOT_SALES
##  Min.   :  1   Length:248232      Min.   :  1.000   Min.   :  1.700
##  1st Qu.: 27   Class :character   1st Qu.:  2.000   1st Qu.:  5.800
##  Median : 52   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56                      Mean   :  1.908   Mean   :  7.308
##  3rd Qu.: 86                      3rd Qu.:  2.000   3rd Qu.:  8.800
##  Max.   :114                      Max.   :200.000   Max.   :650.000
```

After examining the data summary, here is what I've noticed:

- There is 1 year of data: 2018-07-01 to 2019-06-30 ... Great!
- There are no nulls ... Great!
- There are outliers in PROD_QTY and TOT_SALES: the max value of PROD_QTY is 200 but the median is 2 and mean is 1.908. Similar issue in TOT_SALES. Are these outliers related to each other? ... Investigate further!

Print the rows with the Max PROD_QTY:

```
subset(transactionData, PROD_QTY == max(PROD_QTY))
```

```
##             DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 69763 2018-08-19       226         226000 226201        4
## 69764 2019-05-20       226         226000 226210        4
##                             PROD_NAME PROD_QTY TOT_SALES
## 69763 Dorito Corn Chp     Supreme 380g      200       650
## 69764 Dorito Corn Chp     Supreme 380g      200       650
```

This revealed that this large purchase was repeated twice by the same customer. ...Perhaps for a corporate event or other large gathering? Also, I can see that the PROD_QTY outlier (200) is related to the TOT_SALES outlier (650).

Print all transactions from this particular customer:

```
subset(transactionData, LYLTY_CARD_NBR == 226000)
```

```
##             DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR
## 69763 2018-08-19       226         226000 226201        4
## 69764 2019-05-20       226         226000 226210        4
##                             PROD_NAME PROD_QTY TOT_SALES
```

```
## 69763 Dorito Corn Chp     Supreme 380g      200       650
## 69764 Dorito Corn Chp     Supreme 380g      200       650
```

Aha! The 2 outlier transactions were the only transactions from this customer. Due to this,

I will remove these 2 outlier transactions from the dataset to prevent unnecessary skewness in my further analysis.

Remove outlier transactions and recall the summary for this dataset:

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

```
##       DATE              STORE_NBR   LYLTY_CARD_NBR        TXN_ID
##  Min.   :2018-07-01   Min.   :  1   Min.   :   1000   Min.   :      1
##  1st Qu.:2018-09-30   1st Qu.: 70   1st Qu.:  70015   1st Qu.:  67564
##  Median :2018-12-30   Median :130   Median : 130360   Median : 135150
##  Mean   :2018-12-30   Mean   :135   Mean   : 135524   Mean   : 135126
##  3rd Qu.:2019-03-31   3rd Qu.:203   3rd Qu.: 203081   3rd Qu.: 202644
##  Max.   :2019-06-30   Max.   :272   Max.   :2373711   Max.   :2415841
##     PROD_NBR    PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1   Length:248230      Min.   :1.000   Min.   : 1.700
##  1st Qu.: 27   Class :character   1st Qu.:2.000   1st Qu.: 5.800
##  Median : 52   Mode  :character   Median :2.000   Median : 7.400
##  Mean   : 56                      Mean   :1.906   Mean   : 7.303
##  3rd Qu.: 86                      3rd Qu.:2.000   3rd Qu.: 8.800
##  Max.   :114                      Max.   :5.000   Max.   :29.500
```

The max values for PROD+QTY and TOT_SALES now make much more sense, and are in line with the expected results.

CHECK FOR MISSING DATES

Count the number of transactions by date:

```
dateTrans <- transactionData %>%
  count(transactionData$DATE, sort = TRUE)
tibble(dateTrans)
```

```
## # A tibble: 364 x 2
##    `transactionData$DATE`     n
##    <date>                 <int>
##  1 2018-12-24               874
##  2 2018-12-23               856
##  3 2018-12-22               854
##  4 2018-12-19               844
##  5 2018-12-20               811
##  6 2018-12-18               806
##  7 2018-12-21               789
##  8 2019-06-07               761
##  9 2018-09-06               748
## 10 2019-06-14               748
## # ... with 354 more rows
```

There are only 364 rows, but there should be 365 in order to represent a full year. Which one is missing?

At the top of my tibble, I notice that the days with the most transactions are in late December, with December 24 as the top date. My Christmas spirit is first telling me to check if December 25 is missing:

```
subset(transactionData, DATE == "2018-12-25")
```

```
## [1] DATE           STORE_NBR      LYLTY_CARD_NBR TXN_ID         PROD_NBR
## [6] PROD_NAME      PROD_QTY       TOT_SALES
## <0 rows> (or 0-length row.names)
```

There were no transactions on Christmas.

Add the missing date to dateTrans data frame and assign reader-friendly column names

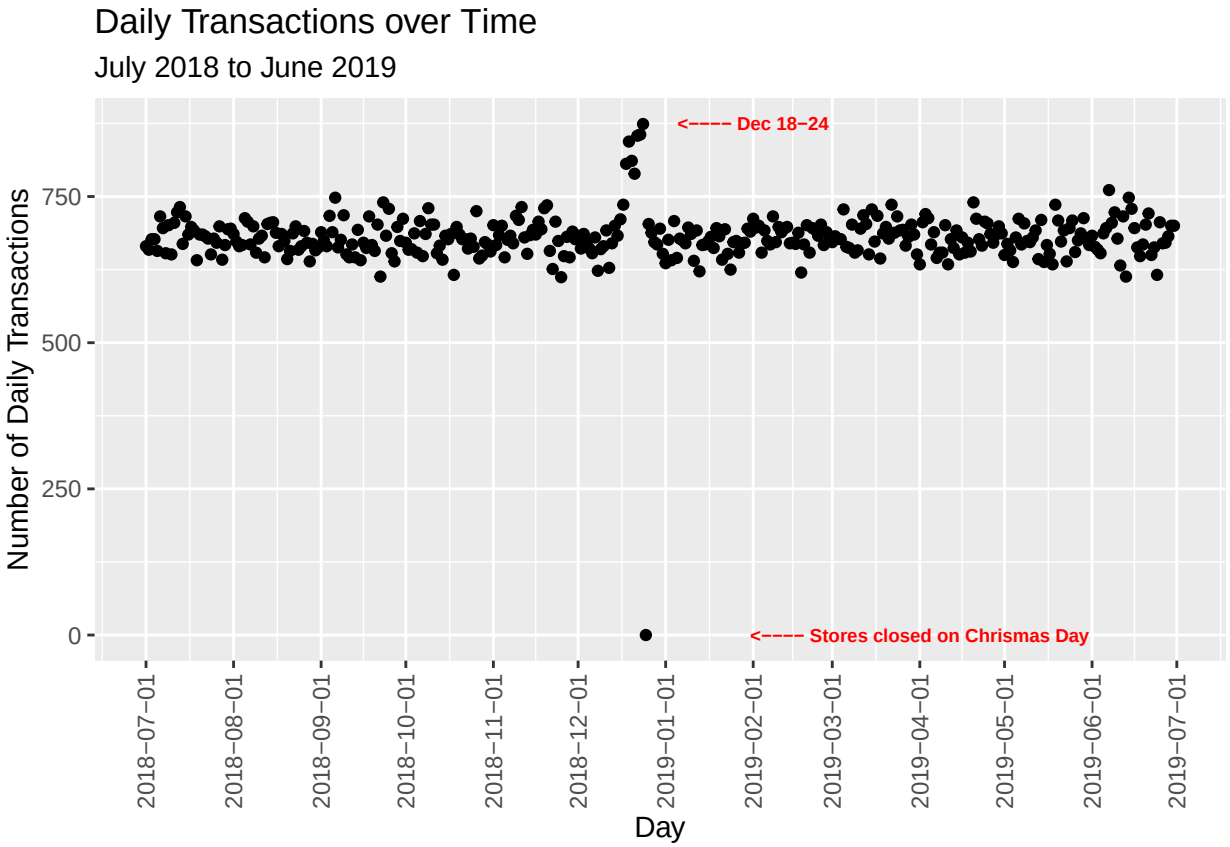
```
dateTrans <- rbind(dateTrans, c("2018-12-25", 0))
colnames(dateTrans) <- c("Date", "NumberOfTransactions")
dateTrans$NumberOfTransactions <- as.numeric(dateTrans$NumberOfTransactions)
head(dateTrans)
```

```
##         Date NumberOfTransactions
## 1 2018-12-24                  874
## 2 2018-12-23                  856
## 3 2018-12-22                  854
## 4 2018-12-19                  844
## 5 2018-12-20                  811
## 6 2018-12-18                  806
```

The chips transactions data frame is now free of nulls, outliers, and missing dates.

At this time, I would also like to visualize the transactions over time:

```
ggplot(dateTrans) +
  geom_point(aes(x = Date, y = NumberOfTransactions)) +
  labs(x = "Day", y = "Number of Daily Transactions",
       title = "Daily Transactions over Time", subtitle = "July 2018 to June 2019") +
  scale_x_date(breaks = "1 month") +
  annotate("text", label = "<---- Dec 18-24", x = as.Date("2019-02-01"), y = 875,
           color = "red", size = 2.5, fontface = "bold") +
  annotate("text", label = "<---- Stores closed on Chrismas Day",
           x = as.Date("2019-04-01"), y = 0, color = "red",
           size = 2.5, fontface = "bold") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```



The graph confirms that chips sales are fairly stable throughout the year, but the week leading up to Christmas there was a sharp spike in sales. There were no sales on Christmas day because the retail stores were closed for the holiday.

Find mean of daily transactions:

```
mean(dateTrans$NumberOfTransactions)
```

```
## [1] 680.0822
```

The average daily transactions for the year is 680.

Find the mean of daily of transactions during the Christmas sales peak:

```
xmasDateTrans <- dateTrans %>%
  filter(Date > "2018-12-17" & Date < "2018-12-25")
mean(xmasDateTrans$NumberOfTransactions)
```

```
## [1] 833.4286
```

Find out how much did sales increase during the Christmas sales peak:

```
mean(xmasDateTrans$NumberOfTransactions) / mean(dateTrans$NumberOfTransactions)
```

```
## [1] 1.225482
```

Sales increased by 22% during the Christmas sales peak.

EXTRACT RELEVANT TRANSACTION DATA

Now I can extract relevant data from this data frame.

Create a Pack Size column by extracting the digits that are in the PROD_NAME column strings:

```
transactionData$PACK_SIZE <- parse_number(transactionData$PROD_NAME)
head(transactionData)
```

```
##         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
## 6 2019-05-16         4           4149   3333       16
##                               PROD_NAME PROD_QTY TOT_SALES PACK_SIZE
## 1   Natural Chip        Compny SeaSalt175g        2       6.0       175
## 2                 CCs Nacho Cheese    175g        3       6.3       175
## 3   Smiths Crinkle Cut  Chips Chicken 170g        2       2.9       170
## 4   Smiths Chip Thinly  S/Cream&Onion 175g        5      15.0       175
## 5 Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13.8       150
## 6 Smiths Crinkle Chips Salt & Vinegar 330g        1       5.7       330
```

Obtain a summary of the new PACK_SIZE column:

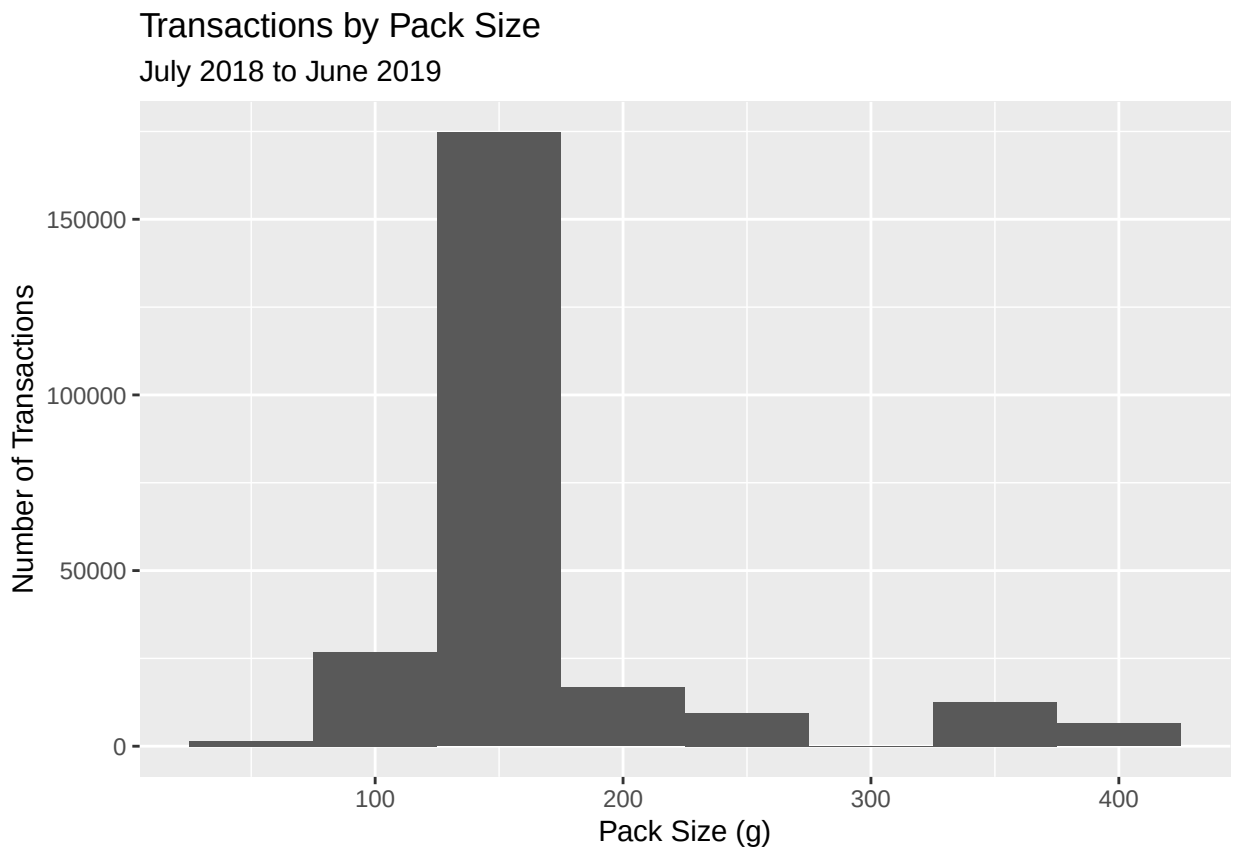
```
summary(transactionData$PACK_SIZE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    70.0   150.0   170.0   175.4   175.0   380.0
```

The minimum chip pack size is 70g, and the maximum chip pack size is 380g.

Plot a histogram of PACK_SIZE:

```
ggplot(transactionData, aes(x = PACK_SIZE)) +
  geom_histogram(binwidth = 50) +
  labs(x = "Pack Size (g)", y = "Number of Transactions",
       title = "Transactions by Pack Size", subtitle = "July 2018 to June 2019")
```



From the histogram, I can see that the majority of chips transactions involved pack sizes between 150-200g, which is consistent with the mean (175.4g) and median (170g).

Create a Brands column by extracting it from the product name:

```
transactionData$BRANDS <- sub(" .*", "", transactionData$PROD_NAME)
head(transactionData)
```

```
##         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
## 6 2019-05-16         4           4149   3333       16
##                                PROD_NAME PROD_QTY TOT_SALES PACK_SIZE  BRANDS
## 1   Natural Chip        Compny SeaSalt175g        2       6.0       175 Natural
## 2                 CCs Nacho Cheese    175g        3       6.3       175     CCs
## 3   Smiths Crinkle Cut  Chips Chicken 170g        2       2.9       170  Smiths
## 4   Smiths Chip Thinly  S/Cream&Onion 175g        5      15.0       175  Smiths
## 5 Kettle Tortilla ChpsHny&Jlpno Chili 150g        3      13.8       150  Kettle
## 6 Smiths Crinkle Chips Salt & Vinegar 330g        1       5.7       330  Smiths
```

How many distinct chips brands are there?

```
n_distinct(transactionData$BRANDS)
```

```
## [1] 28
```

There are 28 distinct brands. Is this right? There may be duplicates or misspellings...

```
uniqueBrands <- data.frame(unique(transactionData$BRANDS))
tibble(uniqueBrands)
```

```
## # A tibble: 28 x 1
##    unique.transactionData.BRANDS.
##    <chr>
##  1 Natural
##  2 CCs
##  3 Smiths
##  4 Kettle
##  5 Grain
##  6 Doritos
##  7 Twisties
##  8 WW
##  9 Thins
## 10 Burger
## # ... with 18 more rows
```

Inspect the tibble above for potential duplicates.

Use the filter in the View pane of UniqueBrands dataframe to search for any potential duplicates

There are several duplicates:

- Dorito and Doritos
- GrnWves and Grain (Waves) (see note below)
- Infzns and Infuzions
- NCC and Natural (Chip Company)
- Red (Rock Deli) and RRD
- Smith and Smiths ... (actually spelled as "Smith's")
- Snbts and Sunbites
- WW and Woolworths

(Grain Waves are actually made by Sunbites, but I have chosen to consider Grain Waves a separate "brand" since they are a distinctly different chip than Sunbites chips.)

Combine duplicate PROD_NAME brands under a unifying BRANDS value:

```
transactionData["BRANDS"][transactionData["BRANDS"] == "Dorito"] <- "Doritos"
transactionData["BRANDS"][transactionData["BRANDS"] == "GrnWves"] <- "Grain Waves"
transactionData["BRANDS"][transactionData["BRANDS"] == "Grain"] <- "Grain Waves"
transactionData["BRANDS"][transactionData["BRANDS"] == "Infzns"] <- "Infuzions"
transactionData["BRANDS"][transactionData["BRANDS"] == "NCC"] <- "Natural Chip Company"
transactionData["BRANDS"][transactionData["BRANDS"] == "Natural"] <- "Natural Chip Company"
transactionData["BRANDS"][transactionData["BRANDS"] == "Red"] <- "Red Rock Deli"
transactionData["BRANDS"][transactionData["BRANDS"] == "RRD"] <- "Red Rock Deli"
transactionData["BRANDS"][transactionData["BRANDS"] == "Smith"] <- "Smith's"
transactionData["BRANDS"][transactionData["BRANDS"] == "Smiths"] <- "Smith's"
transactionData["BRANDS"][transactionData["BRANDS"] == "Snbts"] <- "Sunbites"
transactionData["BRANDS"][transactionData["BRANDS"] == "WW"] <- "Woolworths"
tibble(unique(transactionData$BRANDS))
```

```
## # A tibble: 20 x 1
##    `unique(transactionData$BRANDS)`
##    <chr>
##  1 Natural Chip Company
##  2 CCs
##  3 Smith's
##  4 Kettle
##  5 Grain Waves
##  6 Doritos
##  7 Twisties
##  8 Woolworths
##  9 Thins
## 10 Burger
## 11 Cheezels
## 12 Infuzions
## 13 Red Rock Deli
## 14 Pringles
## 15 Tyrrells
## 16 Cobs
## 17 French
## 18 Tostitos
## 19 Cheetos
## 20 Sunbites
```

Now the chips brands names are cleaned and unified.

Find the most products sold by brand:

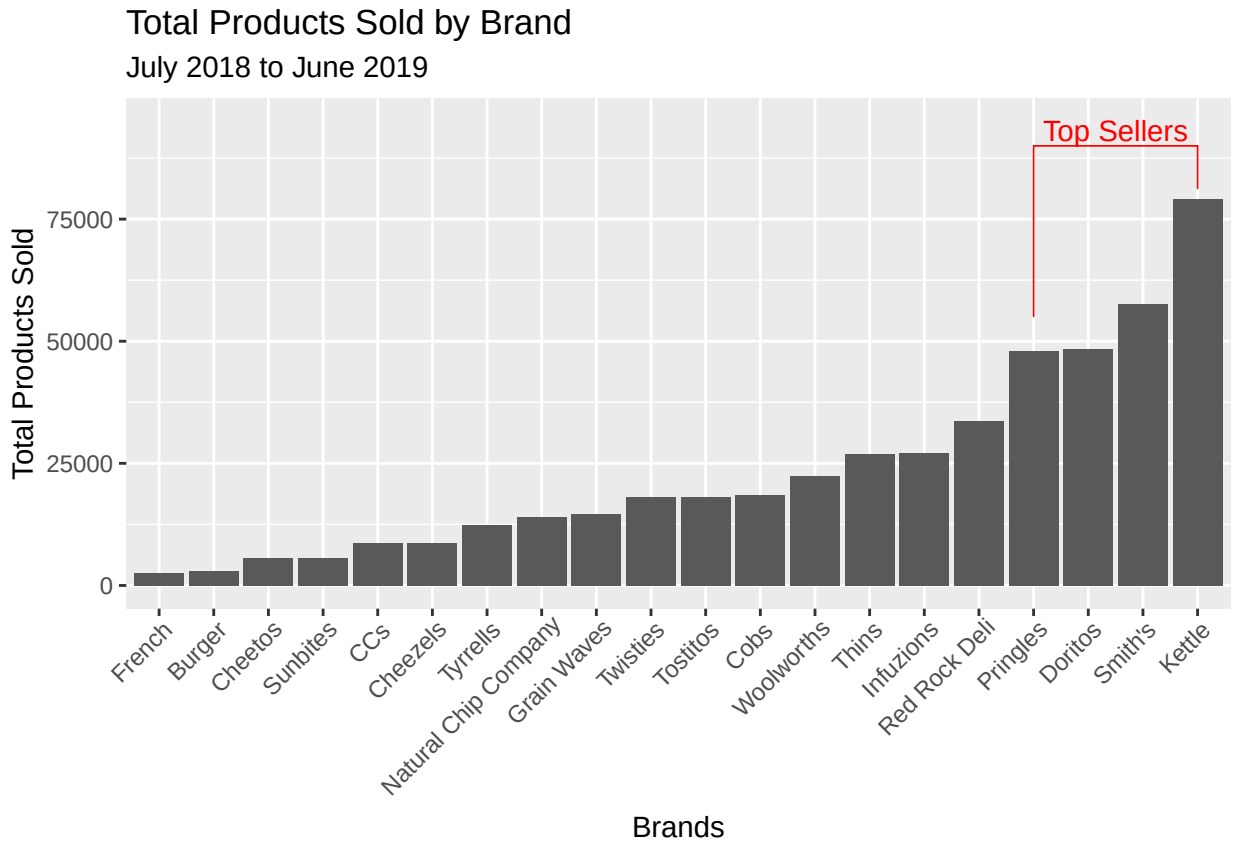
```
brandSales <- transactionData %>%
  group_by(BRANDS) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
brandSales %>%
  arrange(desc(PROD_QTY))
```

```
## # A tibble: 20 x 2
##    BRANDS               PROD_QTY
##    <chr>                   <int>
##  1 Kettle                  79051
##  2 Smith's                 57629
##  3 Doritos                 48331
##  4 Pringles                48019
##  5 Red Rock Deli           33646
##  6 Infuzions               27119
##  7 Thins                   26929
##  8 Woolworths              22333
##  9 Cobs                    18571
## 10 Tostitos                18134
## 11 Twisties                18118
## 12 Grain Waves             14726
## 13 Natural Chip Company    14106
## 14 Tyrrells                12298
## 15 Cheezels                 8747
## 16 CCs                      8609
## 17 Sunbites                 5692
## 18 Cheetos                  5530
## 19 Burger                   2970
```

```
## 20 French                    2643
```

Plot the most products sold by brand:

```
ggplot(brandSales) +
  geom_col(aes(x = reorder(BRANDS, PROD_QTY), y = PROD_QTY)) +
  labs(x = "Brands", y = "Total Products Sold",
       title = "Total Products Sold by Brand", subtitle = "July 2018 to June 2019") +
  geom_bracket(xmin = "Pringles", xmax = "Kettle", y.position = 90000,
               label = "Top Sellers", color = "red", tip.length = c(0.4, 0.1)) +
  coord_cartesian(ylim = c(0, 95000)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.95))
```



Kettle is by far the best selling brand with nearly 80k products sold. Smith's is 2nd best with just under 60k sold. Pringles and Doritos are roughly tied at 3rd with just under 50k products sold.

December 18-24 represent the highest chips sales of the year. Find out which brands sold the best before Christmas:

```
brandSalesXMas <- transactionData %>%
  filter(DATE > "2018-12-17" & DATE < "2018-12-25") %>%
  group_by(BRANDS) %>%
  summarise(XMAS_PROD_QTY = sum(PROD_QTY))
brandSalesXMas %>%
  arrange(desc(XMAS_PROD_QTY))
```

```
## # A tibble: 20 x 2
```

```
##    BRANDS               XMAS_PROD_QTY
##    <chr>                        <int>
##  1 Kettle                        1762
##  2 Smith's                       1292
##  3 Doritos                       1148
##  4 Pringles                      1096
##  5 Red Rock Deli                  806
##  6 Thins                          714
##  7 Infuzions                      618
##  8 Woolworths                     530
##  9 Twisties                       503
## 10 Cobs                           449
## 11 Tostitos                       409
## 12 Grain Waves                    356
## 13 Natural Chip Company           347
## 14 Tyrrells                       290
## 15 CCs                            248
## 16 Cheezels                       219
## 17 Sunbites                       174
## 18 Cheetos                        113
## 19 Burger                          87
## 20 French                          57
```

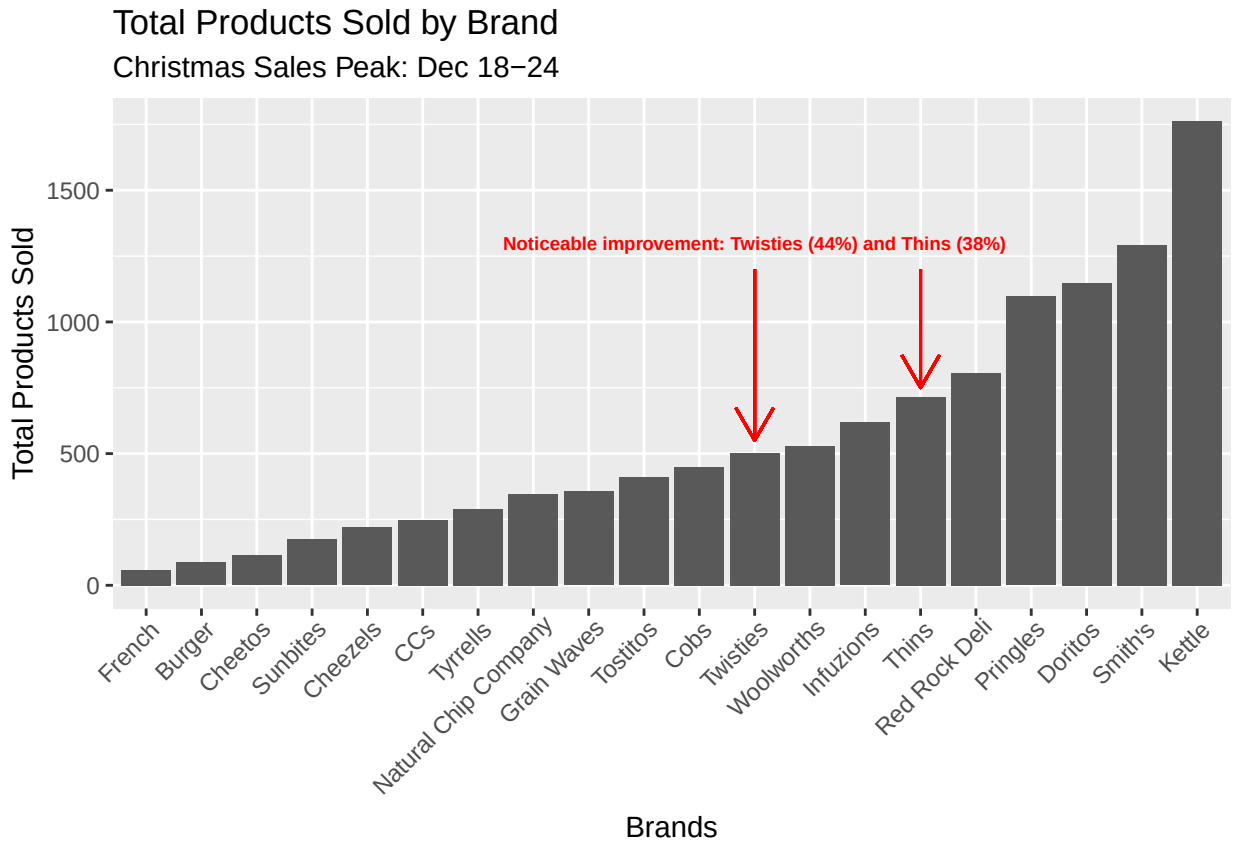
Find out how much Thins sales increased during Christmas sales peak compared to the rest of the year

```
compareBrands <- merge(brandSales, brandSalesXMas, by = "BRANDS")
compareBrands$WEEK_AVG <- round(compareBrands$PROD_QTY / 52)
compareBrands$XMAS_PERC <- compareBrands$XMAS_PROD_QTY / compareBrands$WEEK_AVG
compareBrands %>%
  select(BRANDS, XMAS_PERC) %>%
  arrange(desc(XMAS_PERC))
```

```
##                  BRANDS XMAS_PERC
## 1              Sunbites  1.596330
## 2                Burger  1.526316
## 3                   CCs  1.493976
## 4              Twisties  1.445402
## 5                 Thins  1.378378
## 6              Cheezels  1.303571
## 7  Natural Chip Company  1.280443
## 8           Grain Waves  1.257951
## 9                  Cobs  1.257703
## 10        Red Rock Deli  1.245750
## 11              Doritos  1.235737
## 12           Woolworths  1.235431
## 13             Tyrrells  1.228814
## 14             Pringles  1.187432
## 15            Infuzions  1.183908
## 16             Tostitos  1.171920
## 17              Smith's  1.166065
## 18               Kettle  1.159211
## 19               French  1.117647
## 20              Cheetos  1.066038
```

Plot the most products sold by brand during the sales peak prior to Christmas:

```
ggplot(brandSalesXMas) +
  geom_col(aes(x = reorder(BRANDS, XMAS_PROD_QTY), y = XMAS_PROD_QTY)) +
  labs(x = "Brands", y = "Total Products Sold",
       title = "Total Products Sold by Brand",
       subtitle = "Christmas Sales Peak: Dec 18-24") +
  annotate("text", label = "Noticeable improvement: Twisties (44%) and Thins (38%)",
           x = "Twisties", y = 1300, color = "red", size = 2.5, fontface = "bold") +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.95)) +
  geom_segment(aes(x = "Thins", y = 1200, xend = "Thins", yend = 750),
                 arrow = arrow(length = unit(0.5, "cm")), color = "red") +
  geom_segment(aes(x = "Twisties", y = 1200, xend = "Twisties", yend = 550),
                 arrow = arrow(length = unit(0.5, "cm")), color = "red")
```



The week leading up to Christmas reflects the general year-long brand trend with Kettle, Smith's, Doritos, and Pringles producing overwhelming sales. Thins and Twisties showed noticeable improvement. Sunbites, Burger, and CCs improved more, but their sales quantity is negligable in comparison to Thins and Twisties.

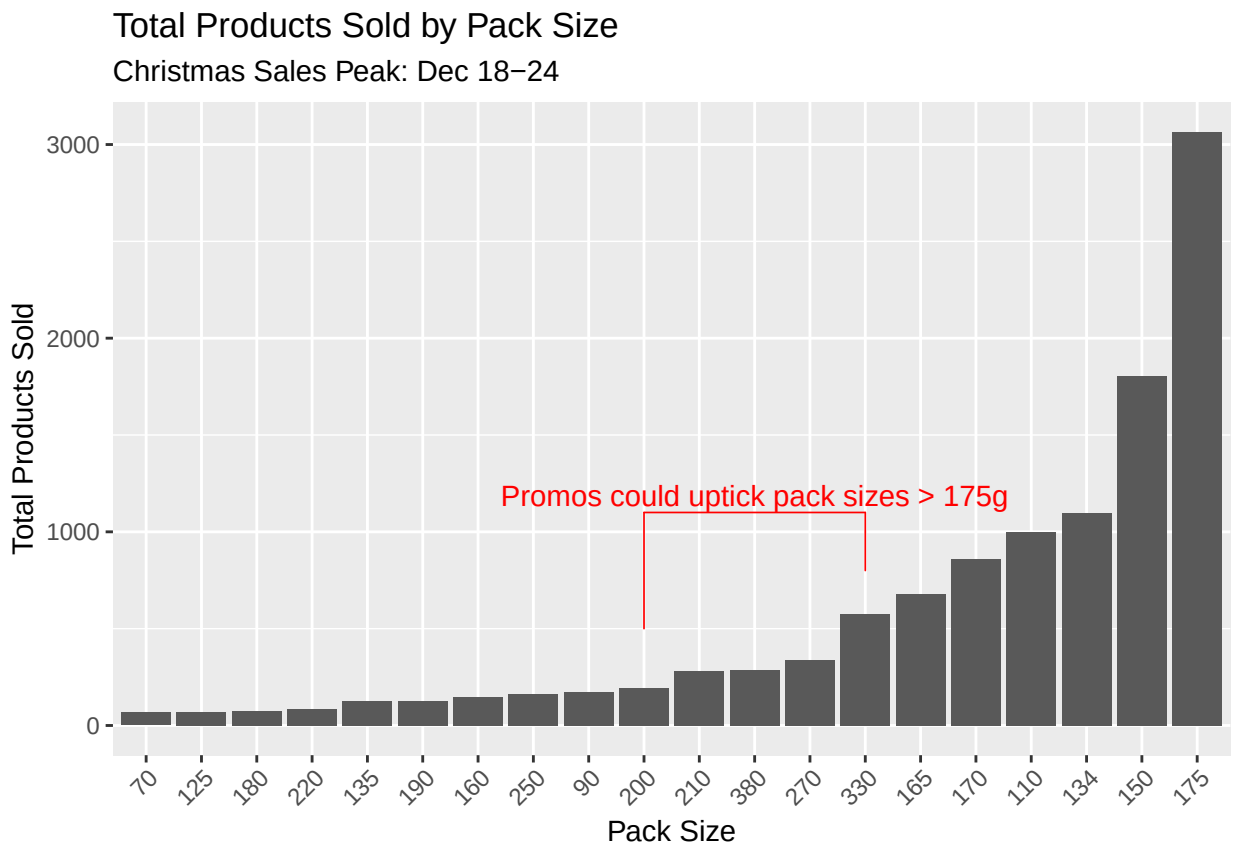
Find out which pack sizes sold the best before Christmas:

```
packSalesXMas <- transactionData %>%
  filter(DATE > "2018-12-17" & DATE < "2018-12-25") %>%
  group_by(PACK_SIZE) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
packSalesXMas %>%
  arrange(desc(PROD_QTY))
```

```
## # A tibble: 20 x 2
##    PACK_SIZE PROD_QTY
##        <dbl>    <int>
##  1       175     3067
##  2       150     1804
##  3       134     1096
##  4       110     1000
##  5       170      860
##  6       165      679
##  7       330      577
##  8       270      338
##  9       380      285
## 10       210      282
## 11       200      193
## 12        90      174
## 13       250      165
## 14       160      148
## 15       190      128
## 16       135      126
## 17       220       87
## 18       180       74
## 19       125       68
## 20        70       67
```

Plot a histogram of PACK_SIZE:

```
ggplot(packSalesXMas) +
  geom_col(aes(x = reorder(PACK_SIZE, PROD_QTY), y = PROD_QTY)) +
  labs(x = "Pack Size", y = "Total Products Sold",
       title = "Total Products Sold by Pack Size",
       subtitle = "Christmas Sales Peak: Dec 18-24") +
  geom_bracket(xmin = "200", xmax = "330", y.position = 1100,
               label = "Promos could uptick pack sizes > 175g",
               color = "red", tip.length = c(0.2, 0.1)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.95))
```



Pack sizes purchased during the Christmas sales peak also reflect the general year-long trend: customers prefer pack sizes between 150-200g. In this case, customers overwhelmingly purchased 175g pack sizes (over 3000 sold in one week), as well as the slightly smaller 150g pack size (1800 sold in one week). Somewhat surprisingly, customers did not prefer packages of a size larger than 200g. Perhaps customers were more interested in purchasing a variety of chips brands during the holidays (in order to satisfy different people's tastes), and therefore stuck with a more mainstream pack size for each brand.

EXPLORE & MERGE CUSTOMER DATA

```
summary(customerData)
```

```
##  LYLTY_CARD_NBR     LIFESTAGE         PREMIUM_CUSTOMER
##  Min.   :   1000   Length:72637       Length:72637
##  1st Qu.:  66202   Class :character   Class :character
##  Median : 134040   Mode  :character   Mode  :character
##  Mean   : 136186
##  3rd Qu.: 203375
##  Max.   :2373711
```

There are no nulls and LYLTY_CARD_NBR is in the same format as in transaction dataframe. This is useful for establishing a relationship between the transaction and customer datasets.

Use a left join to merge transactionData and customerData into one dataframe called "data":

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

```
summary(data)
```

```
##  LYLTY_CARD_NBR         DATE               STORE_NBR       TXN_ID
##  Min.   :   1000   Min.   :2018-07-01   Min.   :  1   Min.   :      1
##  1st Qu.:  70015   1st Qu.:2018-09-30   1st Qu.: 70   1st Qu.:  67564
##  Median : 130360   Median :2018-12-30   Median :130   Median : 135150
##  Mean   : 135524   Mean   :2018-12-30   Mean   :135   Mean   : 135126
##  3rd Qu.: 203081   3rd Qu.:2019-03-31   3rd Qu.:203   3rd Qu.: 202644
##  Max.   :2373711   Max.   :2019-06-30   Max.   :272   Max.   :2415841
##     PROD_NBR    PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1   Length:248230      Min.   :1.000   Min.   : 1.700
##  1st Qu.: 27   Class :character   1st Qu.:2.000   1st Qu.: 5.800
##  Median : 52   Mode  :character   Median :2.000   Median : 7.400
##  Mean   : 56                      Mean   :1.906   Mean   : 7.303
##  3rd Qu.: 86                      3rd Qu.:2.000   3rd Qu.: 8.800
##  Max.   :114                      Max.   :5.000   Max.   :29.500
##    PACK_SIZE        BRANDS           LIFESTAGE         PREMIUM_CUSTOMER
##  Min.   : 70.0   Length:248230      Length:248230      Length:248230
##  1st Qu.:150.0   Class :character   Class :character   Class :character
##  Median :170.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :175.4
##  3rd Qu.:175.0
##  Max.   :380.0
```

There are still no nulls, and there are the same number of rows as in the transactions data frame.

```
head(data)
```

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

It is a successful join. Looks good!

DATA ANALYSIS ON CUSTOMER SEGMENTS

Define some metrics of interest to the client:

1. Who spends the most on chips (total sales), describing customers by lifestage and affluence?
2. How many customers are in each segment?
3. How many chips are bought per customer by segment?
4. What's the average chip price by customer segment?

1. Who spends the most on chips (total sales), describing customers by lifestage and affluence?

Total sales by LIFESTAGE:

```
data %>%
  group_by(LIFESTAGE)  %>%
  summarise(sum_of_sales = sum(TOT_SALES)) %>%
  arrange(desc(sum_of_sales))
```

```
## # A tibble: 7 x 2
##   LIFESTAGE              sum_of_sales
##   <chr>                         <dbl>
## 1 OLDER SINGLES/COUPLES       377381.
## 2 RETIREES                    343782.
## 3 OLDER FAMILIES              330180
## 4 YOUNG FAMILIES              296015.
## 5 YOUNG SINGLES/COUPLES       244478.
## 6 MIDAGE SINGLES/COUPLES      173392.
## 7 NEW FAMILIES                 47510.
```

There are 7 lifestage categories. Older Singles/Couples have the most sales.

Total sales by premium:

```
data %>%
  group_by(PREMIUM_CUSTOMER)  %>%
  summarise(sum_of_sales = sum(TOT_SALES)) %>%
  arrange(desc(sum_of_sales))
```

```
## # A tibble: 3 x 2
##   PREMIUM_CUSTOMER sum_of_sales
##   <chr>                   <dbl>
## 1 Mainstream            703814.
## 2 Budget                633833.
## 3 Premium               475092.
```

There are 3 affluence levels. The Mainstream level has the most sales.

Total sales by customer segment (lifestage + premium as a combined column customer segment):

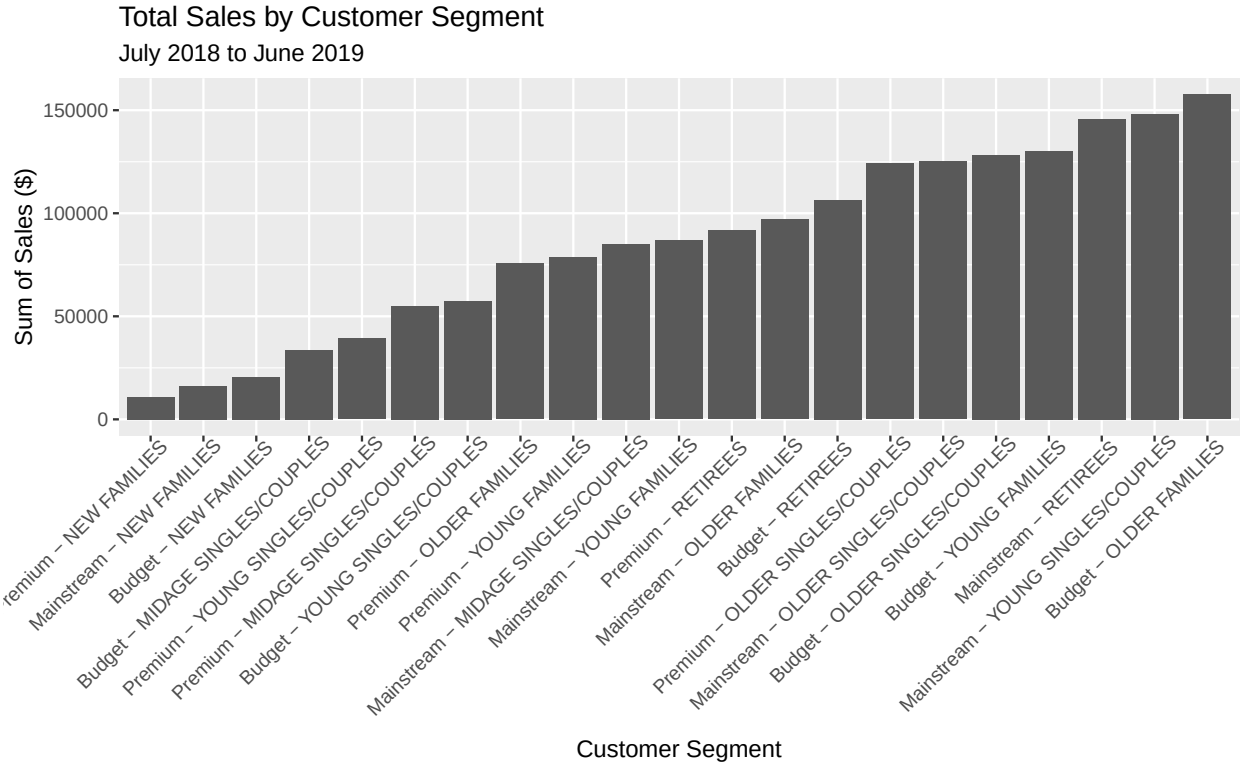
```
data$CUSTOMER_SEGMENT <- paste(data$PREMIUM_CUSTOMER, data$LIFESTAGE, sep=" - ")
salesCustomerSegment <- data %>%
  group_by(CUSTOMER_SEGMENT) %>%
  summarise(SUM_OF_SALES = sum(TOT_SALES))
salesCustomerSegment %>%
  arrange(desc(SUM_OF_SALES))
```

```
## # A tibble: 21 x 2
##    CUSTOMER_SEGMENT                    SUM_OF_SALES
##    <chr>                                      <dbl>
##  1 Budget - OLDER FAMILIES                  157613.
##  2 Mainstream - YOUNG SINGLES/COUPLES       147999.
##  3 Mainstream - RETIREES                    145837.
##  4 Budget - YOUNG FAMILIES                  130352.
##  5 Budget - OLDER SINGLES/COUPLES           128130
##  6 Mainstream - OLDER SINGLES/COUPLES       125178.
##  7 Premium - OLDER SINGLES/COUPLES          124073.
##  8 Budget - RETIREES                        106276
##  9 Mainstream - OLDER FAMILIES               96927.
## 10 Premium - RETIREES                        91669.
## # ... with 11 more rows
```

Top 3 Customer Segments with highest sales: Budget - Older Families, Mainstream - Young Singles/Couples, Mainstream - Retirees.

Visualize the above summary:

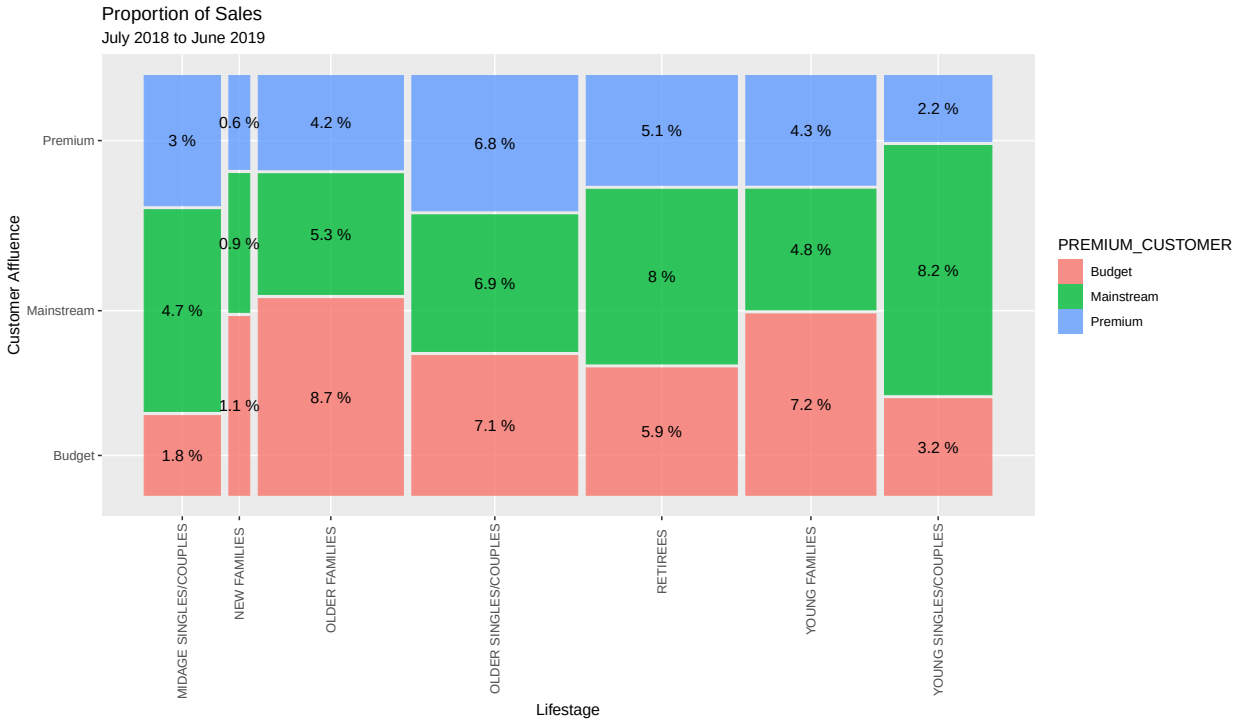
```
ggplot(salesCustomerSegment) +
  geom_col(aes(x = reorder(CUSTOMER_SEGMENT, SUM_OF_SALES), y = SUM_OF_SALES)) +
  labs(x = "Customer Segment", y = "Sum of Sales ($)",
       title = "Total Sales by Customer Segment", subtitle = "July 2018 to June 2019") +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.95))
```



It is easy to see from this column chart that the top 3 customer segments have much higher sales than the other customer segments.

Proportion of sales by customer segment (i.e. lifestage + affluence are separated):

```
salesProportion <- data %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(SALES = sum(TOT_SALES))
p <- ggplot(data = salesProportion) +
  geom_mosaic(aes(weight = SALES, x = product(PREMIUM_CUSTOMER, LIFESTAGE),
                  fill = PREMIUM_CUSTOMER)) +
  labs(x = "Lifestage", y = "Customer Affluence", title = "Proportion of Sales",
       subtitle = "July 2018 to June 2019") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.95))
p + geom_text(data = ggplot_build(p)$data[[1]],
              aes(x = (xmin + xmax)/2 , y = (ymin + ymax)/2,
                  label = as.character(paste(round(.wt/sum(.wt),3)*100, '%'))))
```



The largest proportion of sales are coming from Budget - Older Families (8.7%), Mainstream - Young Singles/Couples (8.2%), and Mainstream - Retirees (8.1%).

Are the higher sales due to there being more customers in these customer segments?

2. How many customers are in each segment?

Count the number of customers by customer segment

```
countCustomerSegment <- data %>%
  select(c(LYLTY_CARD_NBR, CUSTOMER_SEGMENT))
countCustomerSegment <-
  distinct(countCustomerSegment, LYLTY_CARD_NBR, .keep_all = TRUE) %>%
  group_by(CUSTOMER_SEGMENT) %>%
  tally()
countCustomerSegment %>%
  arrange(desc(n))
```

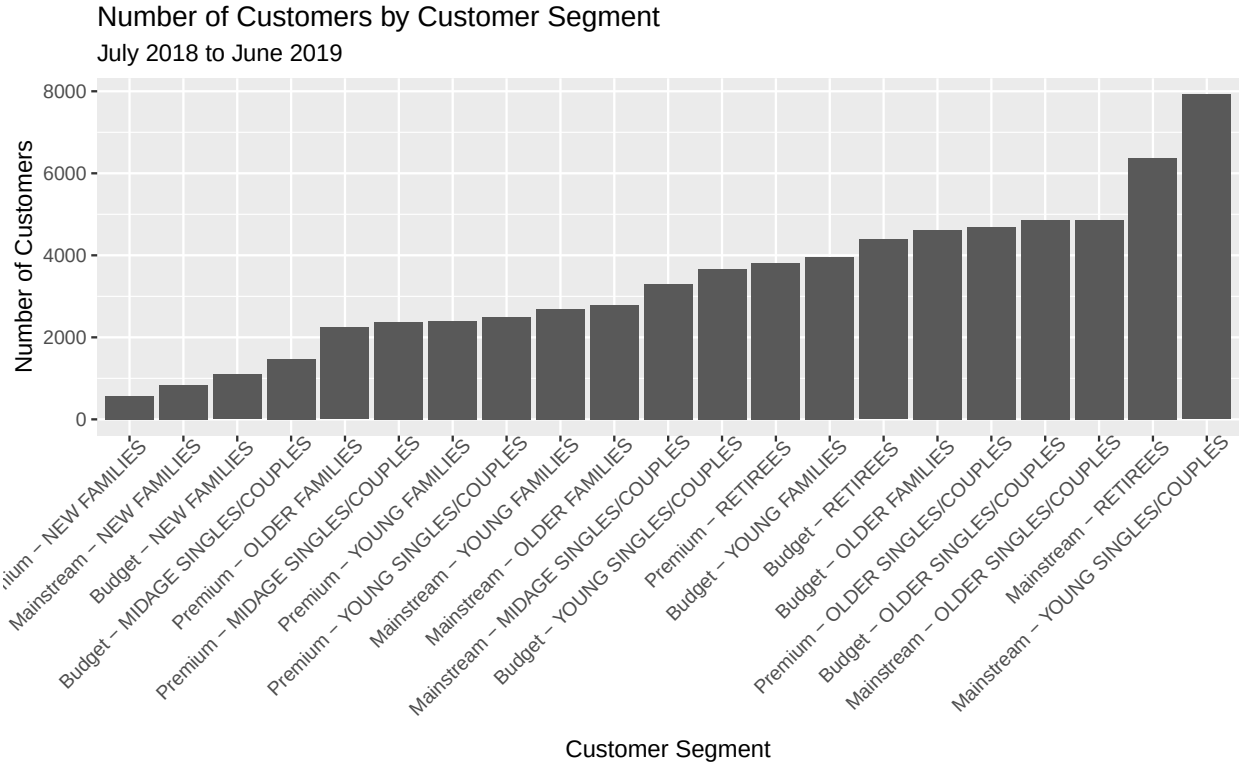
```
## # A tibble: 21 x 2
```

```
##    CUSTOMER_SEGMENT                       n
##    <chr>                              <int>
##  1 Mainstream - YOUNG SINGLES/COUPLES  7921
##  2 Mainstream - RETIREES               6369
##  3 Mainstream - OLDER SINGLES/COUPLES  4866
##  4 Budget - OLDER SINGLES/COUPLES      4856
##  5 Premium - OLDER SINGLES/COUPLES     4692
##  6 Budget - OLDER FAMILIES             4617
##  7 Budget - RETIREES                   4387
##  8 Budget - YOUNG FAMILIES             3959
##  9 Premium - RETIREES                  3818
## 10 Budget - YOUNG SINGLES/COUPLES      3660
## # ... with 11 more rows
```

Top 2 Customer Segments with the most customers: Mainstream - Young Singles/Couples and Mainstream - Retirees. But not Budget - Older Families.

Visualize the above summary:

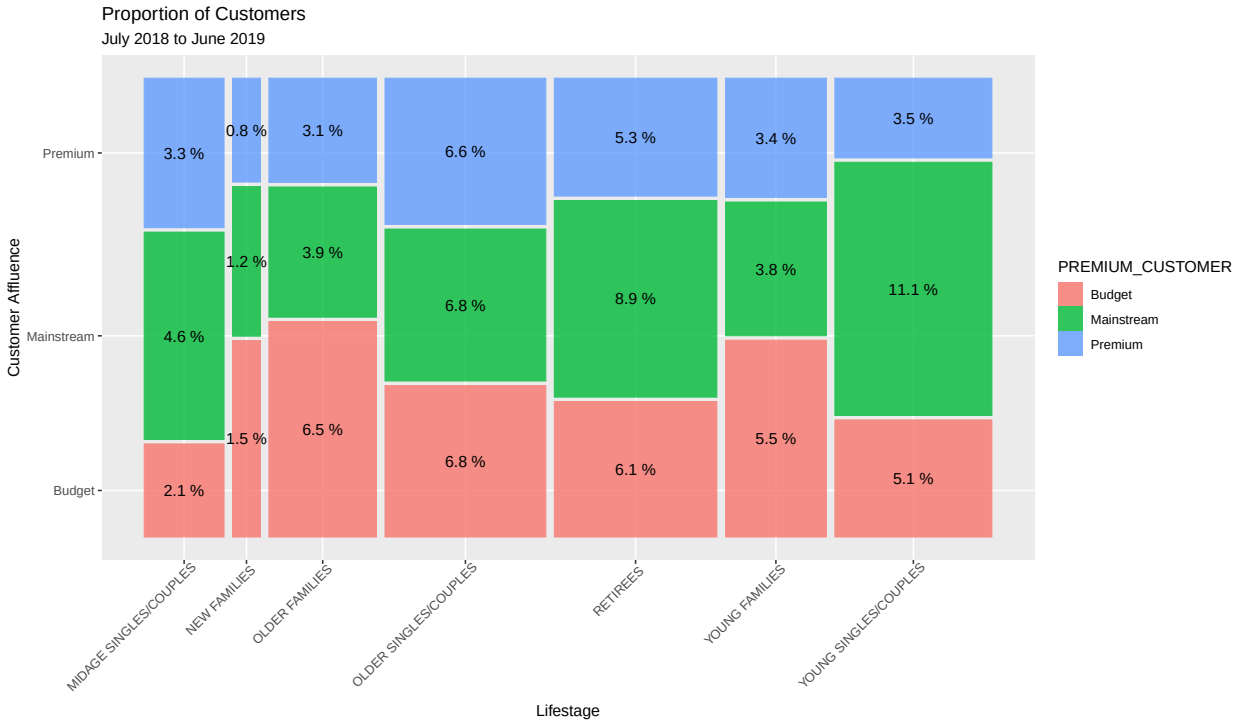
```
ggplot(countCustomerSegment) +
  geom_col(aes(x = reorder(CUSTOMER_SEGMENT, n), y = n)) +
  labs(x = "Customer Segment", y = "Number of Customers",
       title = "Number of Customers by Customer Segment",
       subtitle = "July 2018 to June 2019") +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.95))
```



It is easy to see from this column chart that the top 2 customer segments have much higher number of customers than the other customer segments.

Proportion of customers by customer segment:

```
customerProportion <- data %>%
  select(c(LYLTY_CARD_NBR, LIFESTAGE, PREMIUM_CUSTOMER))
customerProportion <- distinct(customerProportion, LYLTY_CARD_NBR, .keep_all = TRUE) %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  tally()
p <- ggplot(data = customerProportion) +
  geom_mosaic(aes(weight = n, x = product(PREMIUM_CUSTOMER, LIFESTAGE),
                  fill = PREMIUM_CUSTOMER)) +
  labs(x = "Lifestage", y = "Customer Affluence", title = "Proportion of Customers",
       subtitle = "July 2018 to June 2019") +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.95))
p + geom_text(data = ggplot_build(p)$data[[1]],
              aes(x = (xmin + xmax)/2 , y = (ymin + ymax)/2,
                  label = as.character(paste(round(.wt/sum(.wt),3)*100, '%'))))
```



The largest proportion of customers are coming from Mainstream - Young Singles/Couples (11.1%) and Mainstream - Retirees (8.9%). Budget - Older Families underwhelm in this category (6.5%).

The large number of customers in the Mainstream - Young Singles/Couples and Mainstream - Retirees segments contributes to their higher chips sales. This is not true for the Budget - Older Families segment, who incidentally have the highest chips sales. Since the Budget - Older Families segment prefers to purchase cheaper chips, one likely explanation is that they simply purchase more quantity of chips for their potentially larger family size. Older families tend to have larger family sizes (more children, and older children who can consume more chips than younger children). Therefore, is the quantity of chips purchased per transaction a major driver for chips sales?

3. How many chips are bought per customer by segment?

Find the average number of units per customer segment:

```
unitsCustomerSegment <- data %>%
  group_by(LYLTY_CARD_NBR, CUSTOMER_SEGMENT) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
mean(unitsCustomerSegment$PROD_QTY)
```

```
## [1] 6.625702
```

The population avg is 6.6 units (i.e. packages of chips per customer per year).

```
avgUnitsCustomerSegment <- unitsCustomerSegment %>%
  group_by(CUSTOMER_SEGMENT) %>%
  summarise(AVG = mean(PROD_QTY))
avgUnitsCustomerSegment %>%
  arrange(desc(AVG))
```

```
## # A tibble: 21 x 2
##    CUSTOMER_SEGMENT                      AVG
##    <chr>                               <dbl>
##  1 Mainstream - OLDER FAMILIES          9.32
##  2 Budget - OLDER FAMILIES              9.12
##  3 Premium - OLDER FAMILIES             9.11
##  4 Budget - YOUNG FAMILIES              8.77
##  5 Premium - YOUNG FAMILIES             8.74
##  6 Mainstream - YOUNG FAMILIES          8.68
##  7 Premium - OLDER SINGLES/COUPLES      6.80
##  8 Budget - OLDER SINGLES/COUPLES       6.79
##  9 Mainstream - OLDER SINGLES/COUPLES   6.74
## 10 Mainstream - MIDAGE SINGLES/COUPLES  6.46
## # ... with 11 more rows
```

I can see clearly from the tibble above that Older Families dominate the Top 3 in terms of average bags of chips per customer segment, and Young Families round out the Top 6.

Find the average of all Older Families segments:

```
olderFamilies <-
  dplyr::filter(avgUnitsCustomerSegment, grepl("OLDER FAMILIES", CUSTOMER_SEGMENT))
mean(olderFamilies$AVG)
```

```
## [1] 9.183854
```

Older Families average 9.18 packages of chips per customer.

Find the average of all Young Families segments:

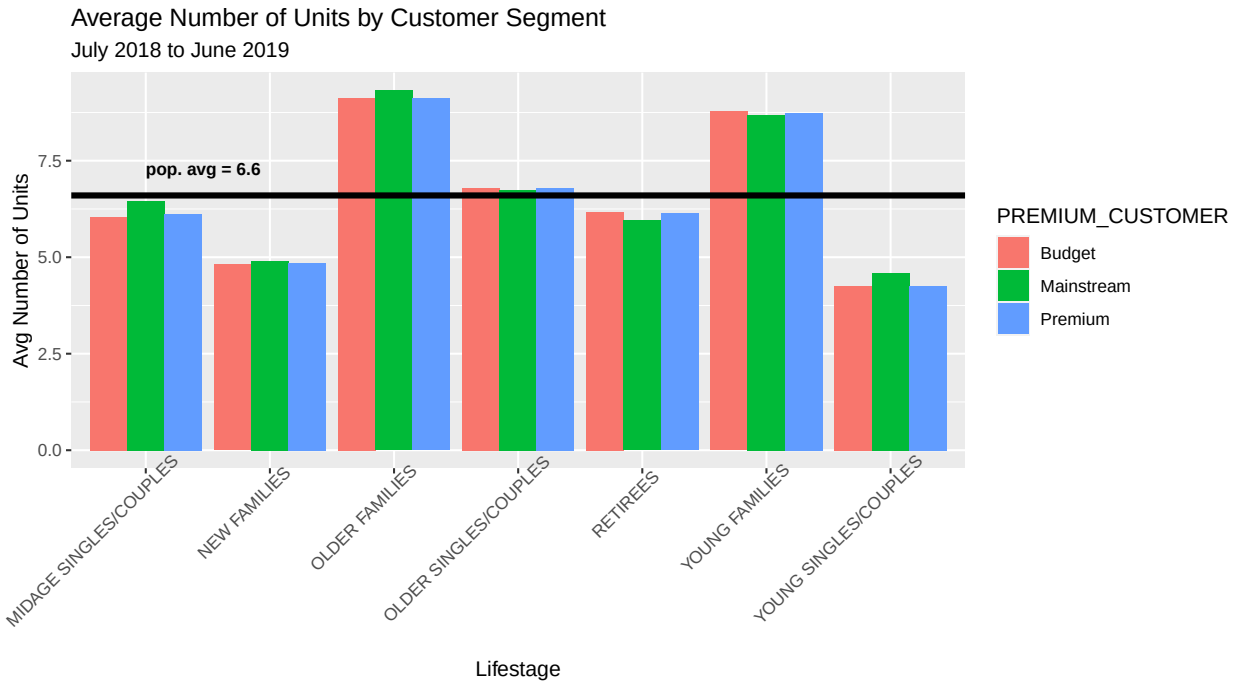
```
youngFamilies <-
  dplyr::filter(avgUnitsCustomerSegment, grepl("YOUNG FAMILIES", CUSTOMER_SEGMENT))
mean(youngFamilies$AVG)
```

```
## [1] 8.73142
```

Young Families average 8.73 packages of chips per customer.

```
unitsCustomerCluster <- data %>%
  group_by(LYLTY_CARD_NBR, LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
avgUnitsCustomerCluster <- unitsCustomerCluster %>%
  group_by(LIFESTAGE, PREMIUM_CUSTOMER) %>%
  summarise(AVG = mean(PROD_QTY))
ggplot(data = avgUnitsCustomerCluster,
```

```
      aes(weight = AVG, x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) +
  geom_bar(position = position_dodge()) +
  labs(x = "Lifestage", y = "Avg Number of Units",
       title = "Average Number of Units by Customer Segment",
       subtitle = "July 2018 to June 2019") +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.85)) +
  geom_hline(yintercept = 6.6, color = "black", size = 1.5)  +
  annotate("text", label = "pop. avg = 6.6", x = "MIDAGE SINGLES/COUPLES", y = 7.3,
           color = "black", size = 3, hjust = "inward", fontface = "bold")
```



Older Families buy the most units per customer (avg 9.18), followed by Young Families (avg 8.73). They are both well above the population average (6.6).

Since families have more people per household, it is logical that they purchase more packages of chips per customer in order to feed more people. Does the price of chips also affect total sales per customer segment?

4. What's the average chip price by customer segment?

Find the average unit price per customer segment:

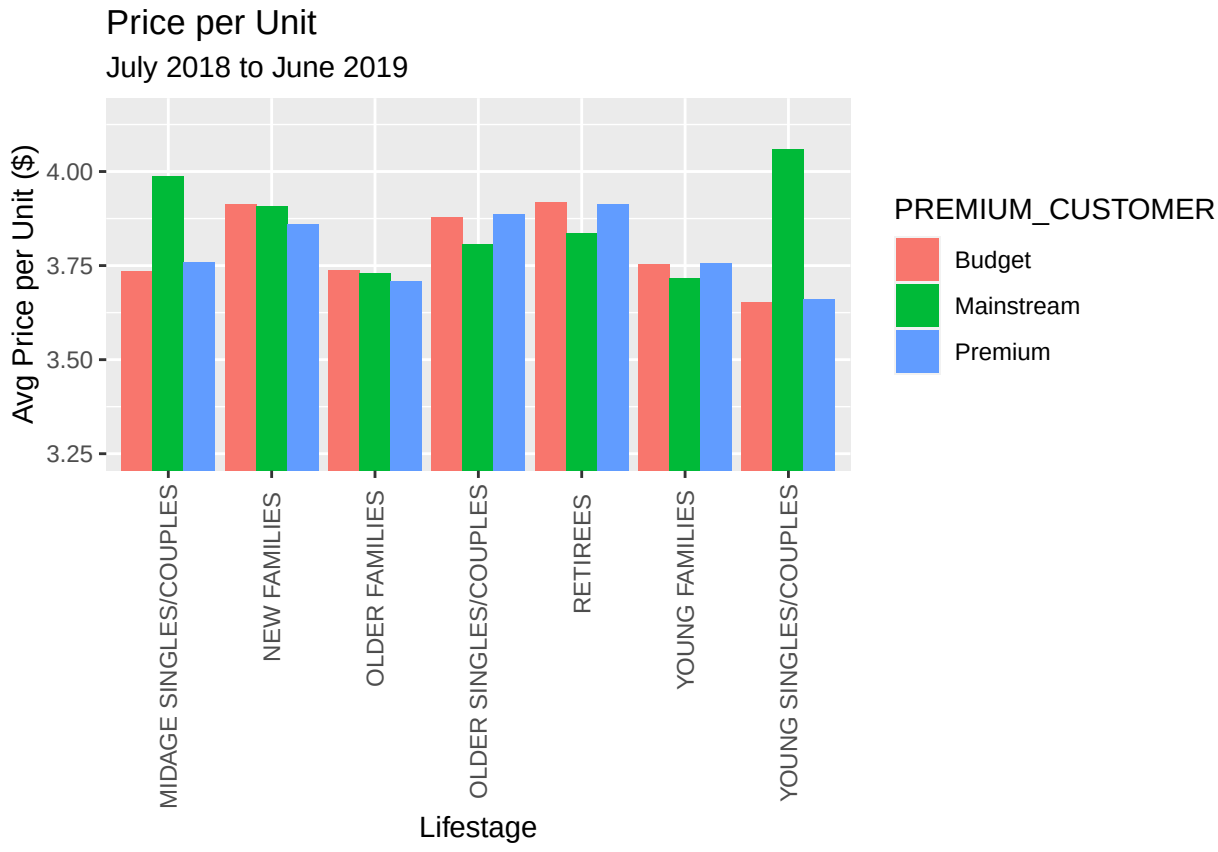
```
priceCustomerCluster <- data %>%
  group_by(PREMIUM_CUSTOMER, LIFESTAGE) %>%
  summarise(AVG = mean(TOT_SALES / PROD_QTY))
priceCustomerCluster %>%
  arrange(desc(AVG))
```

```
## # A tibble: 21 x 3
## # Groups:   PREMIUM_CUSTOMER [3]
##    PREMIUM_CUSTOMER LIFESTAGE                AVG
##    <chr>            <chr>                  <dbl>
##  1 Mainstream       YOUNG SINGLES/COUPLES   4.06
##  2 Mainstream       MIDAGE SINGLES/COUPLES  3.99
```

```
##  3 Budget           RETIREES                3.92
##  4 Budget           NEW FAMILIES            3.91
##  5 Premium          RETIREES                3.91
##  6 Mainstream       NEW FAMILIES            3.91
##  7 Premium          OLDER SINGLES/COUPLES   3.89
##  8 Budget           OLDER SINGLES/COUPLES   3.88
##  9 Premium          NEW FAMILIES            3.86
## 10 Mainstream       RETIREES                3.84
## # ... with 11 more rows
```

Top 2 Customer Segments with the highest average unit price: Mainstream - YOUNG SINGLES/COUPLES and Mainstream - MIDAGE SINGLES/COUPLES. How does this compare to their Budget and Premium counterparts?

```
ggplot(data = priceCustomerCluster,
       aes(weight = AVG, x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) +
  geom_bar(position = position_dodge()) +
  labs(x = "Lifestage", y = "Avg Price per Unit ($)", title = "Price per Unit",
       subtitle = "July 2018 to June 2019") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 0.95)) +
  coord_cartesian(ylim = c(3.25, 4.15))
```



Mainstream - YOUNG SINGLES/COUPLES spent the most amount of money per unit ($4.06) followed by Mainstream - MIDAGE SINGLES/COUPLES ($3.99). Both of these segments vastly outperformed their Budget and Premium counterparts. One likely explanation is that premium chips purchasers may be more likely to purchase healthier snacks in general, thus leading them to only purchase snacks for guests and parties (i.e., they may only purchase cheaper chips brands for other people).

However, the difference in the prices on the graph are not large. The range of the graph is less than a dollar. So, are these average prices statistically different?

PERFORM INDEPENDENT T-TEST BASED ON AVERAGE UNIT PRICE BETWEEN:

"Mainstream - YOUNG and MIDAGE SINGLES/COUPLES" (i.e., the Mainstream segment) vs. "Budget and Premium - YOUNG and MIDAGE SINGLES/COUPLES" (i.e., the non-Mainstream segment)

In this t-test, the null hypothesis is that the Mainstream and non-Mainstream segments will have the same average price per unit. The alternative hyptosis is that these segments will not have same average price per unit.

Create a unit price column for the t-test

```
data$UNIT_PRICE = data$TOT_SALES / data$PROD_QTY
head(data)
```

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

Create values that represent the Mainstream and non-Mainstream segments:

```
target1 <- c("Mainstream - YOUNG SINGLES/COUPLES", "Mainstream - MIDAGE SINGLES/COUPLES")
target2 <- c("Budget - YOUNG SINGLES/COUPLES", "Budget - MIDAGE SINGLES/COUPLES",
```

```
              "Premium - YOUNG SINGLES/COUPLES", "Premium - MIDAGE SINGLES/COUPLES")
```

```
tMainstreamYMASC <- data %>%
  select(c(CUSTOMER_SEGMENT, UNIT_PRICE)) %>%
  filter(CUSTOMER_SEGMENT %in% target1)
mean(tMainstreamYMASC$UNIT_PRICE)
```

```
## [1] 4.033359
```

The average unit price for the Mainstream segment is $4.03.

```
tBudgetPremiumYMASC <- data %>%
  select(c(CUSTOMER_SEGMENT, UNIT_PRICE)) %>%
  filter(CUSTOMER_SEGMENT %in% target2)
mean(tBudgetPremiumYMASC$UNIT_PRICE)
```

```
## [1] 3.699402
```

The average unit price for the non-Mainstream segment is $3.70

The Mainstream segment has a higher average unit price than the non-Mainstream segment.

Conduct the t-test:

```
options(scipen=999) # this disables the scientific notation for clarity
t.test(tMainstreamYMASC$UNIT_PRICE, tBudgetPremiumYMASC$UNIT_PRICE,
       alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  tMainstreamYMASC$UNIT_PRICE and tBudgetPremiumYMASC$UNIT_PRICE
## t = 37.796, df = 55213, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3194235       Inf
## sample estimates:
## mean of x mean of y
##  4.033359  3.699402
```

Since the p-value is less than 0.05, I reject the null hypothesis.

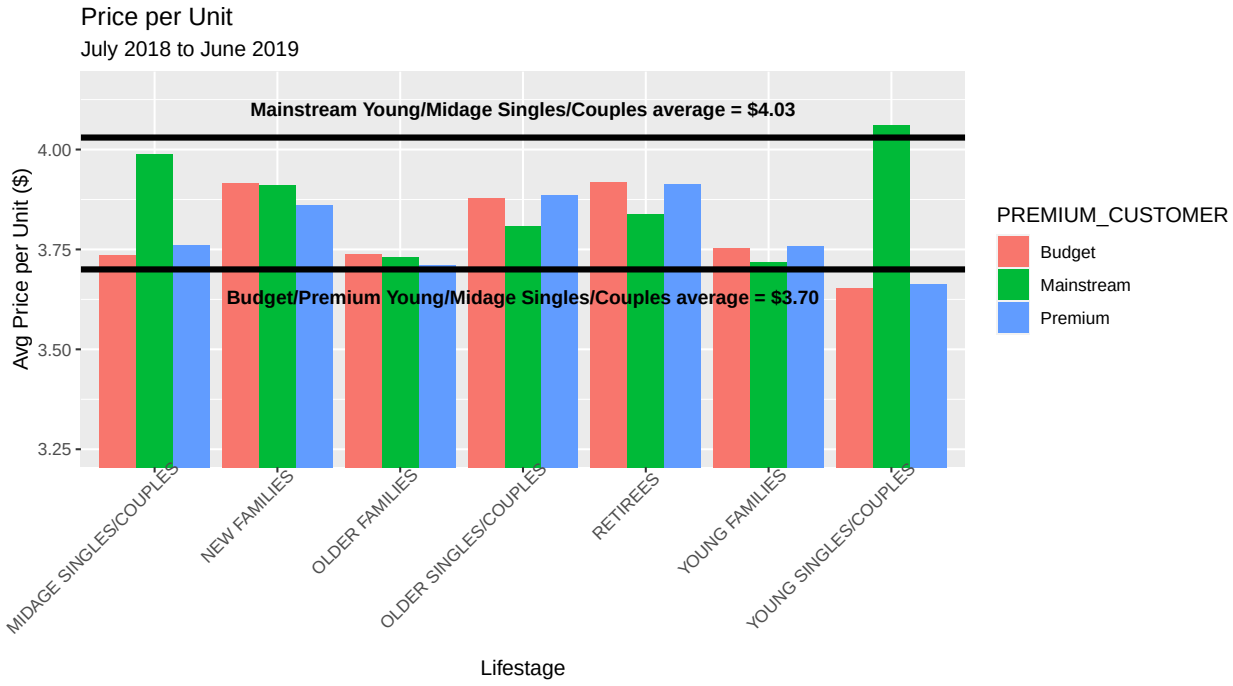
In other words, the mean values of the Mainstream segment's average unit prices and the non-Mainstream segment's average unit prices are significantly different.

In conclusion, the Mainstream segment's average unit prices ($4.03) are statistically and significantly higher than the non-Mainstream segment's average unit prices ($3.70).

Recall the previous plot to visualize this conclusion:

```
ggplot(data = priceCustomerCluster,
       aes(weight = AVG, x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) +
  geom_bar(position = position_dodge()) +
  labs(x = "Lifestage", y = "Avg Price per Unit ($)", title = "Price per Unit",
       subtitle = "July 2018 to June 2019") +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.9)) +
  coord_cartesian(ylim = c(3.25, 4.15)) +
  geom_hline(yintercept = 4.03, color = "black", size = 1.5) +
```

```
  geom_hline(yintercept = 3.70, color = "black", size = 1.5) +
  annotate("text", label = "Mainstream Young/Midage Singles/Couples average = $4.03",
           x = "OLDER SINGLES/COUPLES", y = 4.10,
           color = "black", size = 3.5, fontface = "bold") +
  annotate("text", label = "Budget/Premium Young/Midage Singles/Couples average = $3.70",
           x = "OLDER SINGLES/COUPLES", y = 3.63,
           color = "black", size = 3.5, fontface = "bold")
```



AFFINITY ANALYSIS ON TARGET CUSTOMER SEGMENT

The Mainstream - YOUNG SINGLES/COUPLES customer segment is a significant customer segment as it is near the top of many of my analyses thus far, including the t-test conclusion. I will dive deeper into this customer segment through brand and pack size affinity analysis (i.e., brand preference and pack size preference).

Do Mainstream - YOUNG SINGLES/COUPLES prefer certain brands more than others?

For this analysis, the Mainstream - YOUNG SINGLES/COUPLES is the "Target Segment", and all other customer segments are the "Other Segments".

Create values that represent that Mainstream - YOUNG SINGLES/COUPLES segment:

```
target3 <- c("Mainstream - YOUNG SINGLES/COUPLES")
```

Find the product quantity by brand purchased by the Target Segment:

```
brandsMainstreamYSC <- data %>%
  filter(CUSTOMER_SEGMENT %in% target3) %>%
  group_by(BRANDS) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
tibble(brandsMainstreamYSC) %>%
  arrange(desc(PROD_QTY))
```

```
## # A tibble: 20 x 2
##    BRANDS                PROD_QTY
##    <chr>                    <int>
##  1 Kettle                    7172
##  2 Doritos                   4447
##  3 Pringles                  4326
##  4 Smith's                   3479
##  5 Infuzions                 2343
##  6 Thins                     2187
##  7 Red Rock Deli             1753
##  8 Twisties                  1673
##  9 Tostitos                  1645
## 10 Cobs                      1617
## 11 Grain Waves               1185
## 12 Tyrrells                  1143
## 13 Woolworths                 873
## 14 Natural Chip Company       710
## 15 Cheezels                   651
## 16 CCs                        405
## 17 Cheetos                    291
## 18 Sunbites                   230
## 19 French                     143
## 20 Burger                     106
```

Find the product quantity by brand purchased by the Other Segments:

```
brandsOtherSegments <- data %>%
  filter(!(CUSTOMER_SEGMENT %in% target3)) %>%
  group_by(BRANDS) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
tibble(brandsOtherSegments) %>%
  arrange(desc(PROD_QTY))
```

```
## # A tibble: 20 x 2
##    BRANDS                PROD_QTY
##    <chr>                    <int>
##  1 Kettle                   71879
##  2 Smith's                  54150
##  3 Doritos                  43884
##  4 Pringles                 43693
##  5 Red Rock Deli            31893
##  6 Infuzions                24776
##  7 Thins                    24742
##  8 Woolworths               21460
##  9 Cobs                     16954
## 10 Tostitos                 16489
## 11 Twisties                 16445
## 12 Grain Waves              13541
## 13 Natural Chip Company     13396
## 14 Tyrrells                 11155
## 15 CCs                       8204
## 16 Cheezels                  8096
## 17 Sunbites                  5462
## 18 Cheetos                   5239
## 19 Burger                    2864
```

```
## 20 French                    2500
```

For the Target Segment, calculate the proportion of each brand purchased against all other brands purchased:

```
quantity_brandsMainstreamYSC <- brandsMainstreamYSC %>%
  summarise(BRANDS, TARGET_SEGMENT = PROD_QTY / sum(PROD_QTY))
tibble(quantity_brandsMainstreamYSC) %>%
  arrange(desc(TARGET_SEGMENT))
```

```
## # A tibble: 20 x 2
##    BRANDS               TARGET_SEGMENT
##    <chr>                         <dbl>
##  1 Kettle                       0.197
##  2 Doritos                      0.122
##  3 Pringles                     0.119
##  4 Smith's                      0.0956
##  5 Infuzions                    0.0644
##  6 Thins                        0.0601
##  7 Red Rock Deli                0.0482
##  8 Twisties                     0.0460
##  9 Tostitos                     0.0452
## 10 Cobs                         0.0444
## 11 Grain Waves                  0.0326
## 12 Tyrrells                     0.0314
## 13 Woolworths                   0.0240
## 14 Natural Chip Company         0.0195
## 15 Cheezels                     0.0179
## 16 CCs                          0.0111
## 17 Cheetos                      0.00800
## 18 Sunbites                     0.00632
## 19 French                       0.00393
## 20 Burger                       0.00291
```

For the Other Segments, calculate the proportion of each brand purchased against all other brands purchased:

```
quantity_brandsOtherSegments <- brandsOtherSegments %>%
  summarise(BRANDS, OTHER_SEGMENTS = PROD_QTY / sum(PROD_QTY))
tibble(quantity_brandsOtherSegments) %>%
  arrange(desc(OTHER_SEGMENTS))
```

```
## # A tibble: 20 x 2
##    BRANDS               OTHER_SEGMENTS
##    <chr>                         <dbl>
##  1 Kettle                       0.165
##  2 Smith's                      0.124
##  3 Doritos                      0.100
##  4 Pringles                     0.100
##  5 Red Rock Deli                0.0730
##  6 Infuzions                    0.0567
##  7 Thins                        0.0566
##  8 Woolworths                   0.0491
##  9 Cobs                         0.0388
## 10 Tostitos                     0.0377
## 11 Twisties                     0.0376
## 12 Grain Waves                  0.0310
## 13 Natural Chip Company         0.0307
```

```
## 14 Tyrrells                   0.0255
## 15 CCs                        0.0188
## 16 Cheezels                   0.0185
## 17 Sunbites                   0.0125
## 18 Cheetos                    0.0120
## 19 Burger                     0.00656
## 20 French                     0.00572
```

Merge these proportions into one data frame and calculate the brand affinity:

```
brand_affinity <- merge(quantity_brandsMainstreamYSC, quantity_brandsOtherSegments) %>%
  summarise(BRANDS, TARGET_SEGMENT, OTHER_SEGMENTS,
            AFFINITY_TO_BRAND = TARGET_SEGMENT / OTHER_SEGMENTS)
brand_affinity %>%
  arrange(desc(AFFINITY_TO_BRAND))
```

```
##                   BRANDS TARGET_SEGMENT OTHER_SEGMENTS AFFINITY_TO_BRAND
## 1               Tyrrells    0.031419225    0.025536717         1.2303549
## 2               Twisties    0.045988070    0.037646913         1.2215628
## 3                Doritos    0.122240853    0.100461973         1.2167873
## 4                 Kettle    0.197146706    0.164549862         1.1980971
## 5               Tostitos    0.045218395    0.037747641         1.1979131
## 6               Pringles    0.118914759    0.100024724         1.1888537
## 7                   Cobs    0.044448720    0.038812148         1.1452270
## 8              Infuzions    0.064405289    0.056718755         1.1355201
## 9                  Thins    0.060117101    0.056640920         1.0613722
## 10           Grain Waves    0.032573738    0.030998897         1.0508031
## 11              Cheezels    0.017894939    0.018533865         0.9655266
## 12               Smith's    0.095632095    0.123963537         0.7714534
## 13                French    0.003930839    0.005723155         0.6868308
## 14               Cheetos    0.007999120    0.011993444         0.6669578
## 15         Red Rock Deli    0.048187141    0.073011433         0.6599945
## 16 Natural Chip Company    0.019516754    0.030666954         0.6364099
## 17                   CCs    0.011132796    0.018781105         0.5927658
## 18              Sunbites    0.006322329    0.012503949         0.5056266
## 19            Woolworths    0.023997361    0.049127562         0.4884704
## 20                Burger    0.002913769    0.006556446         0.4444128
```

Mainstream - YOUNG SINGLES/COUPLES are 23% more likely to purchase Tyrrells chips compared to the rest of the population

Mainstream - YOUNG SINGLES/COUPLES are 56% less likely to purchase Burger Rings compared to the rest of the population

Do Mainstream - YOUNG SINGLES/COUPLES prefer certain pack sizes more than others?

I will perform similar steps here as the previous brand affinity analysis, and use the same Target Segment and Other Segments.

Find the product quantity by pack size purchased by the Target Segment:

```
packMainstreamYSC <- data %>%
  filter(CUSTOMER_SEGMENT %in% target3) %>%
  group_by(PACK_SIZE) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
tibble(packMainstreamYSC) %>%
  arrange(desc(PROD_QTY))
```

```
## # A tibble: 20 x 2
##    PACK_SIZE PROD_QTY
##        <dbl>    <int>
##  1       175     9237
##  2       150     5863
##  3       134     4326
##  4       110     3850
##  5       170     2926
##  6       330     2220
##  7       165     2016
##  8       380     1165
##  9       270     1153
## 10       210     1055
## 11       135      535
## 12       250      520
## 13       200      325
## 14       190      271
## 15       160      232
## 16        90      230
## 17       180      130
## 18        70      110
## 19       125      109
## 20       220      106
```

Find the product quantity by pack size purchased by the Other Segments:

```
packOtherSegments <- data %>%
  filter(!(CUSTOMER_SEGMENT %in% target3)) %>%
  group_by(PACK_SIZE) %>%
  summarise(PROD_QTY = sum(PROD_QTY))
tibble(packOtherSegments) %>%
  arrange(desc(PROD_QTY))
```

```
## # A tibble: 20 x 2
##    PACK_SIZE PROD_QTY
##        <dbl>    <int>
##  1       175   117230
##  2       150    73601
##  3       134    43693
##  4       110    38985
##  5       170    35162
##  6       165    27035
##  7       330    21779
##  8       380    11108
##  9       210    10907
## 10       270    10896
## 11       200     8100
## 12       135     5677
## 13       250     5549
## 14        90     5462
## 15       190     5402
## 16       160     5372
## 17       220     2864
## 18        70     2745
## 19       180     2634
```

```
## 20       125     2621
```

For the Target Segment, calculate the proportion of each pack size purchased against all other pack sizes purchased:

```
quantity_packMainstreamYSC <- packMainstreamYSC %>%
  summarise(PACK_SIZE, TARGET_SEGMENT = PROD_QTY / sum(PROD_QTY))
tibble(quantity_packMainstreamYSC) %>%
  arrange(desc(TARGET_SEGMENT))
```

```
## # A tibble: 20 x 2
##    PACK_SIZE TARGET_SEGMENT
##        <dbl>          <dbl>
##  1       175        0.254
##  2       150        0.161
##  3       134        0.119
##  4       110        0.106
##  5       170        0.0804
##  6       330        0.0610
##  7       165        0.0554
##  8       380        0.0320
##  9       270        0.0317
## 10       210        0.0290
## 11       135        0.0147
## 12       250        0.0143
## 13       200        0.00893
## 14       190        0.00745
## 15       160        0.00638
## 16        90        0.00632
## 17       180        0.00357
## 18        70        0.00302
## 19       125        0.00300
## 20       220        0.00291
```

For the Other Segments, calculate the proportion of each pack size purchased against all other pack sizes purchased:

```
quantity_packOtherSegments <- packOtherSegments %>%
  summarise(PACK_SIZE, OTHER_SEGMENTS = PROD_QTY / sum(PROD_QTY))
tibble(quantity_packOtherSegments) %>%
  arrange(desc(OTHER_SEGMENTS))
```

```
## # A tibble: 20 x 2
##    PACK_SIZE OTHER_SEGMENTS
##        <dbl>          <dbl>
##  1       175        0.268
##  2       150        0.168
##  3       134        0.100
##  4       110        0.0892
##  5       170        0.0805
##  6       165        0.0619
##  7       330        0.0499
##  8       380        0.0254
##  9       210        0.0250
## 10       270        0.0249
## 11       200        0.0185
```

```
## 12       135        0.0130
## 13       250        0.0127
## 14        90        0.0125
## 15       190        0.0124
## 16       160        0.0123
## 17       220        0.00656
## 18        70        0.00628
## 19       180        0.00603
## 20       125        0.00600
```

Merge these proportions into one data frame and calculate the pack size affinity:

```
pack_affinity <- merge(quantity_packMainstreamYSC, quantity_packOtherSegments) %>%
  summarise(PACK_SIZE, TARGET_SEGMENT, OTHER_SEGMENTS,
            AFFINITY_TO_PACK_SIZE = TARGET_SEGMENT / OTHER_SEGMENTS)
pack_affinity %>%
  arrange(desc(AFFINITY_TO_PACK_SIZE))
```

```
##    PACK_SIZE TARGET_SEGMENT OTHER_SEGMENTS AFFINITY_TO_PACK_SIZE
## 1        270    0.031694109    0.024943799             1.2706208
## 2        380    0.032023970    0.025429122             1.2593423
## 3        330    0.061024217    0.049857837             1.2239644
## 4        134    0.118914759    0.100024724             1.1888537
## 5        110    0.105830287    0.089246879             1.1858150
## 6        210    0.029000247    0.024968981             1.1614510
## 7        135    0.014706287    0.012996140             1.1315888
## 8        250    0.014293961    0.012703115             1.1252328
## 9        170    0.080431018    0.080495030             0.9992048
## 10       150    0.161164408    0.168491972             0.9565109
## 11       175    0.253910223    0.268370183             0.9461194
## 12       165    0.055416586    0.061890198             0.8954017
## 13       190    0.007449353    0.012366593             0.6023771
## 14       180    0.003573490    0.006029916             0.5926269
## 15       160    0.006377306    0.012297915             0.5185680
## 16        90    0.006322329    0.012503949             0.5056266
## 17       125    0.002996234    0.006000156             0.4993594
## 18       200    0.008933726    0.018543022             0.4817837
## 19        70    0.003023722    0.006284024             0.4811761
## 20       220    0.002913769    0.006556446             0.4444128
```

Mainstream - YOUNG SINGLES/COUPLES are 27% more likely to purchase a 270g pack of chips compared to the rest of the population

WHAT BRANDS SELL THIS PACK SIZE?

Find out if there is correlation between brand affinity and pack size affinity:

```
data %>%
  group_by(PROD_NAME) %>%
  distinct(PACK_SIZE) %>%
  filter(PACK_SIZE == 270)
```

```
## # A tibble: 2 x 2
## # Groups:   PROD_NAME [2]
##   PROD_NAME                PACK_SIZE
##   <chr>                        <dbl>
## 1 Twisties Cheese     270g      270
```

```
## 2 Twisties Chicken270g           270
```

Twisties are the only brand offering 270g packs and so this may instead be reflecting a higher likelihood of purchasing Twisties. Mainstream - YOUNG SINGLES/COUPLES are 22% more likely to purchase Twisties, so this may explain a correlation between brand affinity and pack size affinity.

EXPORT MERGED DATA AS CSV FOR TASK 2 AND 3

```
write.csv(data,"C:/Users/garci/OneDrive/Desktop/Data Analysis Education/Forage Virtual Internships/Quant
```

INSIGHTS:

- Sales increased by 22% during the Christmas sales peak.
- The majority of chips transactions involved pack sizes between 150-200g.
- Kettle is by far the best selling brand with nearly 80k products sold. Smith's is 2nd best with just under 60k sold. Pringles and Doritos are roughly tied at 3rd with just under 50k products sold.
- The week leading up to Christmas reflects the general year-long brand trend with Kettle, Smith's, Doritos, and Pringles producing overwhelming sales. Thins and Twisties showed the most improvement.
- Pack sizes purchased during the Christmas sales peak also reflect the general year-long trend: customers prefer pack sizes between 150-200g. In this case, customers overwhelmingly purchased 175g pack sizes (over 3000 sold in one week), as well as the slightly smaller 150g pack size (1800 sold in one week). Somewhat surprisingly, customers did not prefer packages of a size larger than 200g. Perhaps customers were more interested in purchasing a variety of chips brands during the holidays (in order to satisfy different people's tastes), and therefore stuck with a more mainstream pack size for each brand.
- The largest proportion of sales are coming from Budget - Older Families (8.7%), Mainstream - Young Singles/Couples (8.2%), and Mainstream - Retirees (8.1%).
- The large number of customers in the Mainstream - Young Singles/Couples and Mainstream - Retirees segments contributes to their higher chips sales. This is not true for the Budget - Older Families segment, who incidentally have the highest chips sales. Since the Budget - Older Families segment prefers to purchase cheaper chips, one likely explanation is that they simply purchase more quantity of chips for their potentially larger family size. Older families tend to have larger family sizes (more children, and older children who can consume more chips than younger children).
- Older Families buy the most units per customer (avg 9.18), followed by Young Families (avg 8.73). They are both well above the population average (6.6). Since families have more people per household, it is logical that they purchase more packages of chips per customer in order to feed more people.
- Mainstream - YOUNG SINGLES/COUPLES spent the most amount of money per unit ($4.06) followed by Mainstream - MIDAGE SINGLES/COUPLES ($3.99). Both of these segments vastly outperformed their Budget and Premium counterparts. One likely explanation is that premium chips purchasers may be more likely to purchase healthier snacks in general, thus leading them to only purchase snacks for guests and parties (i.e., they may only purchase cheaper chips brands for other people).
- The Mainstream segment's average unit prices ($4.03) are statistically and significantly higher than the non-Mainstream segment's average unit prices ($3.70).
- Target customer segment: Mainstream - YOUNG SINGLES/COUPLES (they are near the top of majority of analyses).
- Mainstream - YOUNG SINGLES/COUPLES are 23% more likely to purchase Tyrrells chips compared to the rest of the population.

- Mainstream - YOUNG SINGLES/COUPLES are 56% less likely to purchase Burger Rings compared to the rest of the population.
- Mainstream - YOUNG SINGLES/COUPLES are 27% more likely to purchase a 270g pack of chips compared to the rest of the population
- Twisties are the only brand offering 270g packs and so this may instead be reflecting a higher likelihood of purchasing Twisties. Mainstream - YOUNG SINGLES/COUPLES are 22% more likely to purchase Twisties, so this may explain a correlation between brand affinity and pack size affinity.

COMMERCIAL RECOMMENDATIONS:

- Kettles, Smith's, Pringles, and Doritos are by far the best selling brands. These should be kept well in stock year-round.
- Thins and Twisties show a slight sales improvement prior to Christmas. This could be useful for promoting them during the holiday season.
- Since customers continue to purchase the standard 150-200g pack sizes during the Christmas sales peak, there could be more promotional offers on pack sizes of 200g+.
- Twisties could also be off-located in other parts of the store that mainstream young singles/couples are likely to be shopping.
- Similarly, other top brands with large pack size like Doritos can be offlocated in same location as Twisties to increase sales.
- It's possible that large pack size brands like Smiths and Cheezels could see an uptick if also offlocated to similar parts of the store.