Data Preparation and Customer analytics

Aremu Oluwasegun

2024-01-13

Loading Required Libraries and Datasets

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.1
```

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.2.1
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.3
```

```
library(stringr)
```

```
## Warning: package 'stringr' was built under R version 4.2.1
```

```
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 4.2.3

##
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
purchase_behaviour <- read_csv("C:/Data analysis/Projects Notebook amd markdowns/Hands-on project/Datas
```

```
## Rows: 72637 Columns: 3

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): LIFESTAGE, PREMIUM_CUSTOMER
## dbl (1): LYLTY_CARD_NBR
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
transaction_data <- read_excel("C:/Data analysis/Projects Notebook amd markdowns/Hands-on project/Datas
View(transaction_data)
```

Exploratory Analysis

We start by checking the formats of the cloumns in the trasaction dataset

```
str(transaction_data)
```

```
## tibble [264,836 x 8] (S3: tbl_df/tbl/data.frame)
##  $ DATE          : num [1:264836] 43390 43599 43605 43329 43330 ...
##  $ STORE_NBR     : num [1:264836] 1 1 1 2 2 4 4 4 5 7 ...
##  $ LYLTY_CARD_NBR: num [1:264836] 1000 1307 1343 2373 2426 ...
##  $ TXN_ID        : num [1:264836] 1 348 383 974 1038 ...
##  $ PROD_NBR      : num [1:264836] 5 66 61 69 108 57 16 24 42 52 ...
##  $ PROD_NAME     : chr [1:264836] "Natural Chip        Compny SeaSalt175g" "CCs Nacho Cheese    175g
##  $ PROD_QTY      : num [1:264836] 2 3 2 5 3 1 1 1 1 2 ...
##  $ TOT_SALES     : num [1:264836] 6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...
```

```
### And also previewing the irst 10 rows of the dataset
head(transaction_data)
```

```
## # A tibble: 6 x 8
##    DATE STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME    PROD_QTY TOT_SALES
##   <dbl>     <dbl>          <dbl>  <dbl>    <dbl> <chr>           <dbl>     <dbl>
## 1 43390         1           1000      1        5 Natural Chi~        2       6
## 2 43599         1           1307    348       66 CCs Nacho C~        3       6.3
## 3 43605         1           1343    383       61 Smiths Crin~        2       2.9
## 4 43329         2           2373    974       69 Smiths Chip~        5      15
## 5 43330         2           2426   1038      108 Kettle Tort~        3      13.8
## 6 43604         4           4074   2982       57 Old El Paso~        1       5.1
```

We can see that the date is in Numeric datatype format. We then convert it to date format.

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

We should also examine the product name to ensure we are looking at the right products.

```
transaction_data %>%
  group_by(PROD_NAME) %>%
  summarise(count = n())
```

```
## # A tibble: 114 x 2
##    PROD_NAME                                count
##    <chr>                                    <int>
##  1 Burger Rings 220g                         1564
##  2 CCs Nacho Cheese    175g                  1498
##  3 CCs Original 175g                         1514
##  4 CCs Tasty Cheese    175g                  1539
##  5 Cheetos Chs & Bacon Balls 190g            1479
##  6 Cheetos Puffs 165g                        1448
##  7 Cheezels Cheese 330g                      3149
##  8 Cheezels Cheese Box 125g                  1454
##  9 Cobs Popd Sea Salt  Chips 110g            3265
## 10 Cobs Popd Sour Crm  &Chives Chips 110g    3159
## # i 104 more rows
```

Now , we examine the product name to make sure we are looking at chips.

```
unique_words <- unique(transaction_data$PROD_NAME)
wrapped_unique_words <- str_wrap(unique_words)
productWords <- data.table(unlist(strsplit(wrapped_unique_words, split = " ")))
setnames(productWords, 'words')
```

As our interest lies only in chips, we will remove digits, and special character form the Product names

```
productWords_1 <- productWords[!grepl("&",productWords$words),]
 productwords_new<- productWords_1[!grepl("[0-9]",productWords_1$words),]
```

Now we count the frequency of each words and then we sort.

```
productwords_new %>%
  group_by(words) %>%
  summarise(No_of_times = n()) %>%
  arrange(desc(No_of_times))
```

```
## # A tibble: 171 x 2
##    words    No_of_times
##    <chr>          <int>
##  1 Chips             21
##  2 Smiths            16
##  3 Crinkle           14
##  4 Cut               14
```

```
##   5 Kettle            13
##   6 Cheese            12
##   7 Salt              12
##   8 Original          10
##   9 Chip               9
##  10 Doritos            9
## # i 161 more rows
```

There are salsa products in the dataset, and we have to remove them since we are working with chips.

```
transaction_data<- transaction_data[!grepl("Salsa",transaction_data$PROD_NAME), ]
```

Now we want to check for Outliers in the data using the summary function

```
summary(transaction_data)
```

```
##       DATE              STORE_NBR     LYLTY_CARD_NBR       TXN_ID
##  Min.   :2018-07-01   Min.   :  1.0   Min.   :   1000   Min.   :      1
##  1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  70015   1st Qu.:  67569
##  Median :2018-12-30   Median :130.0   Median : 130367   Median : 135183
##  Mean   :2018-12-30   Mean   :135.1   Mean   : 135531   Mean   : 135131
##  3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203084   3rd Qu.: 202654
##  Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841
##     PROD_NBR       PROD_NAME            PROD_QTY          TOT_SALES
##  Min.   :  1.00   Length:246742      Min.   :  1.000   Min.   :  1.700
##  1st Qu.: 26.00   Class :character   1st Qu.:  2.000   1st Qu.:  5.800
##  Median : 53.00   Mode  :character   Median :  2.000   Median :  7.400
##  Mean   : 56.35                      Mean   :  1.908   Mean   :  7.321
##  3rd Qu.: 87.00                      3rd Qu.:  2.000   3rd Qu.:  8.800
##  Max.   :114.00                      Max.   :200.000   Max.   :650.000
```

There are no nulls in the dataset but there seem to be an outlier in the PROD_QTY where there was a purchase of 200 packets of chip by a customer. ### Filtering the data to find the Outlier

```
transaction_data %>%
  filter(PROD_QTY == 200)
```

```
## # A tibble: 2 x 8
##   DATE       STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME         PROD_QTY
##   <date>         <dbl>          <dbl>  <dbl>    <dbl> <chr>                <dbl>
## 1 2018-08-19       226         226000 226201        4 Dorito Corn Chp ~      200
## 2 2019-05-20       226         226000 226210        4 Dorito Corn Chp ~      200
## # i 1 more variable: TOT_SALES <dbl>
```

There are 2 transactions where the customer bought 200 packets , now we check if there other transaction by this customer

```
transaction_data %>%
  filter(LYLTY_CARD_NBR == 226000)
```

```
## # A tibble: 2 x 8
##   DATE       STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME         PROD_QTY
##   <date>         <dbl>          <dbl>  <dbl>    <dbl> <chr>                <dbl>
## 1 2018-08-19       226         226000 226201        4 Dorito Corn Chp ~      200
## 2 2019-05-20       226         226000 226210        4 Dorito Corn Chp ~      200
## # i 1 more variable: TOT_SALES <dbl>
```

We can see that mo other transaction were made by this customer except for these 2, We assume the customer might be buying for commercial purposes. So we remove these transaction from further analysis. ### Remmoving the Outlier

```
transaction_data<- transaction_data[!grepl("22600",transaction_data$LYLTY_CARD_NBR), ]
## Re-examing the dataset
summary(transaction_data)
```

```
##       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.:  70014   1st Qu.:  67557
##  Median :2018-12-30   Median :130   Median : 130362   Median : 135159
##  Mean   :2018-12-30   Mean   :135   Mean   : 135515   Mean   : 135115
##  3rd Qu.:2019-03-31   3rd Qu.:203   3rd Qu.: 203076   3rd Qu.: 202621
##  Max.   :2019-06-30   Max.   :272   Max.   :2373711   Max.   :2415841
##     PROD_NBR        PROD_NAME            PROD_QTY       TOT_SALES
##  Min.   :  1.00   Length:246698      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
```

Analysing date column

Now we want to check the transaction lines over to observe if there are any missing values.

```
transaction_data %>%
  group_by(DATE) %>%
  summarise(count = n())
```

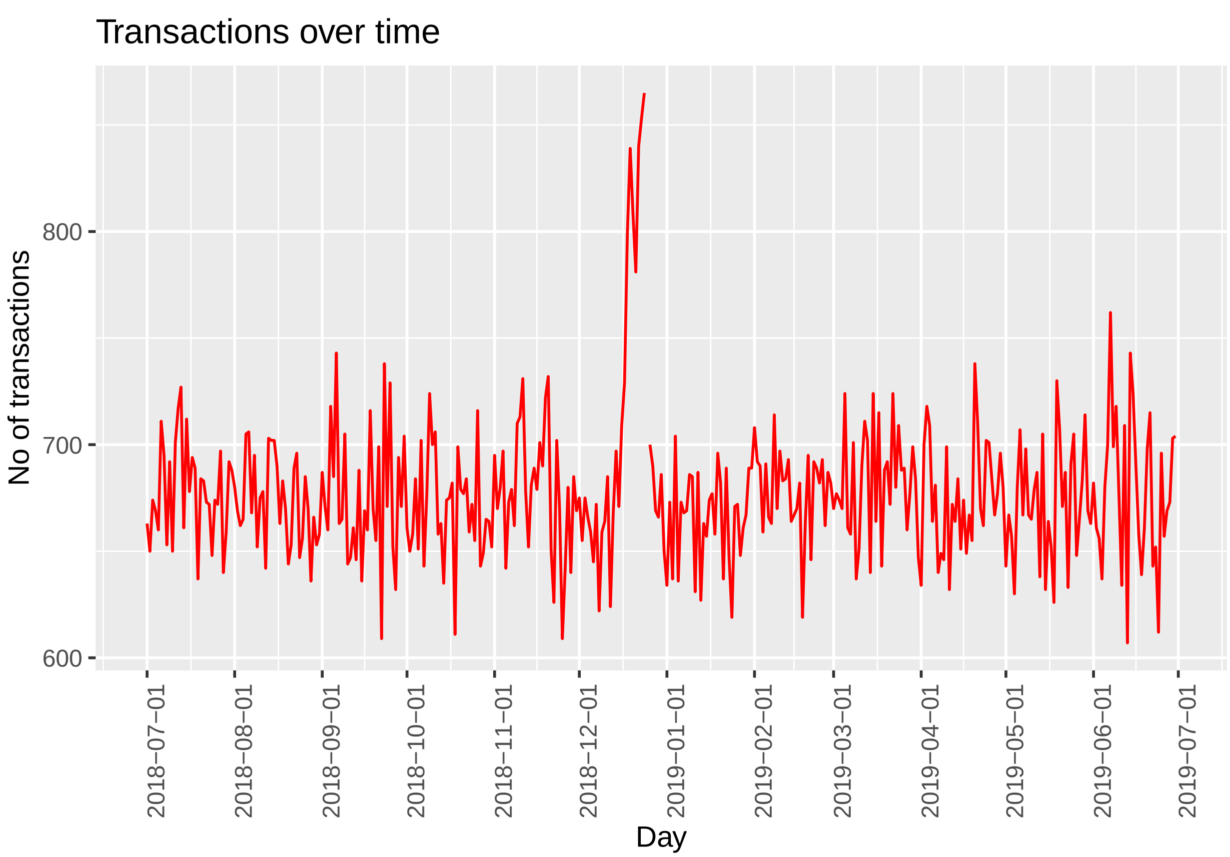
```
## # A tibble: 364 x 2
##    DATE       count
##    <date>     <int>
##  1 2018-07-01   663
##  2 2018-07-02   650
##  3 2018-07-03   674
##  4 2018-07-04   669
##  5 2018-07-05   660
##  6 2018-07-06   711
##  7 2018-07-07   695
##  8 2018-07-08   653
##  9 2018-07-09   692
## 10 2018-07-10   650
## # i 354 more rows
```

There are only 364 rows in the result, meaning one day is missing. To find this missing date, we create a sequence of date from 1st of july, 2018 to 30th of june, 2019 and add this to the dataset to find the missing date.

```
## Creating the sequence of date
all_dates <- data.table(seq(as.Date("2018/07/01"), as.Date("2019/06/30"), by = "day"))
setnames(all_dates, "DATE")
## Joining to the transaction table
transaction_dates <- transaction_data %>%
  group_by(DATE) %>%
  summarise(count = n())
transaction_by_day <- left_join(all_dates,transaction_dates, by = "DATE")
```

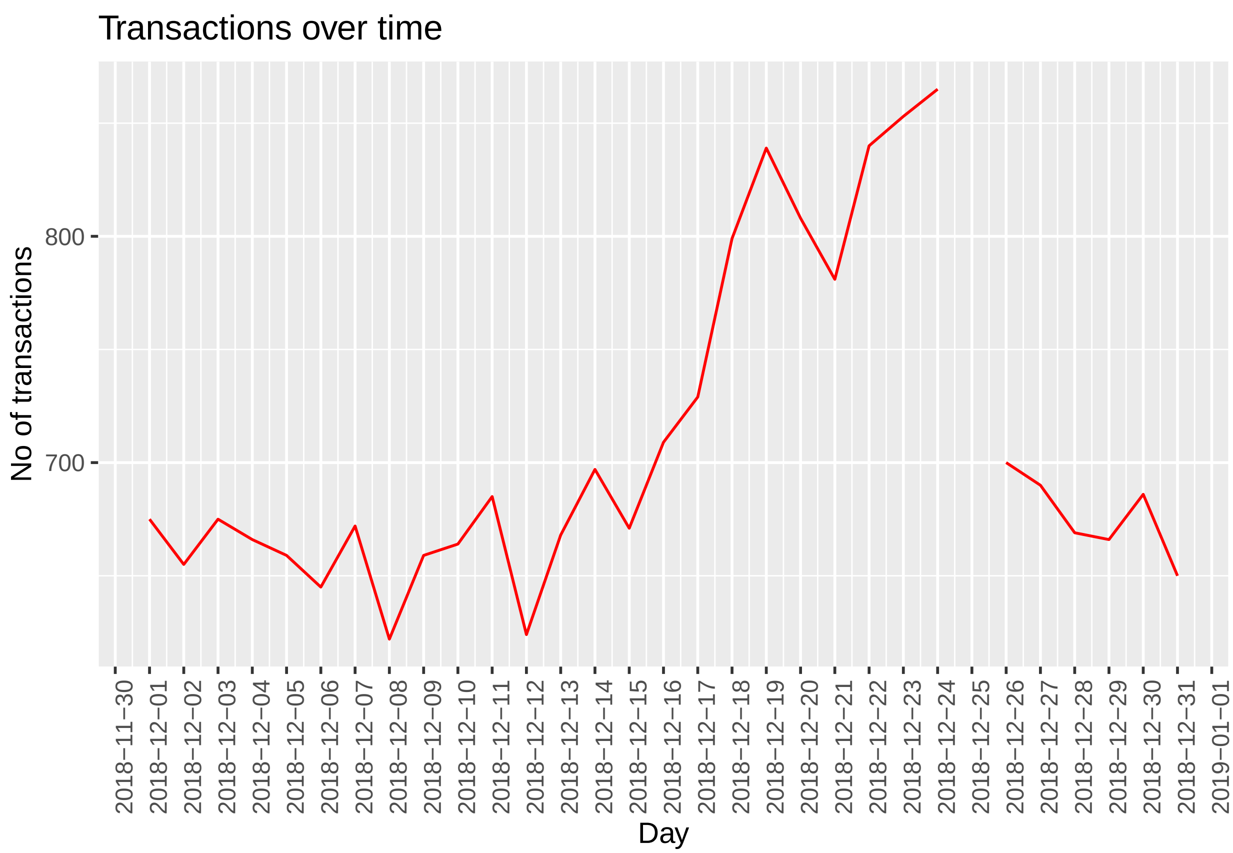
Now we create a line plot to check the missing date

```
##line plot
ggplot(transaction_by_day, aes(x = DATE, y = count)) +
 geom_line(col="red") +
 labs(x = "Day", y = "No of transactions", title = "Transactions over time") +
  scale_x_date(breaks = "1 month") +
 theme(axis.text.x = element_text(angle = 90, hjust = 0.5))
```



We can see that there is an increase in price in December and a break in late December.

```r
## Filtering to look at December by day
Dec_tran <- transaction_by_day %>%
  filter(month(DATE) == 12)
## Plotting December Transactions by day
ggplot(Dec_tran, aes(x = DATE, y = count)) +
 geom_line(col="red") +
 labs(x = "Day", y = "No of transactions", title = "Transactions over time") +
  scale_x_date(breaks = "1 day") +
 theme(axis.text.x = element_text(angle = 90, hjust = 0.5))
```



We can see that the increase in purchases are the days leading up to Christmas and there was no record for Christmas because of the Christmas holiday.

Now that we have proven that there no more Outlier in the data, we can move to create other features such as pack size or brand name from PROD_NAME. ## Creating Pack sizes column

```r
## We start by creating pack sizes from PROD_NAME
transaction_data <- transaction_data %>%
  mutate(pack_size = parse_number(transaction_data$PROD_NAME))
## Ordering from to check if it makes sense
transaction_data %>%
  arrange(pack_size)
```

```
## # A tibble: 246,698 x 9
##    DATE       STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME        PROD_QTY
##    <date>         <dbl>          <dbl>  <dbl>    <dbl> <chr>               <dbl>
```

```
##   1 2019-03-15         5          5091   4905       38 Infuzions Mango~        2
##   2 2018-08-01        10         10192  10175       38 Infuzions Mango~        2
##   3 2019-01-17        39         39134  35441       38 Infuzions Mango~        2
##   4 2018-08-21        39         39144  35500       38 Infuzions Mango~        2
##   5 2018-10-08        48         48009  43109       38 Infuzions Mango~        2
##   6 2019-02-05        55         55114  49137       38 Infuzions Mango~        2
##   7 2019-03-21        97         97089  96849       38 Infuzions Mango~        2
##   8 2019-04-10       128        128105 131225       38 Infuzions Mango~        2
##   9 2019-02-11       129        129136 133036       38 Infuzions Mango~        2
##  10 2019-06-22       129        129184 133337       38 Infuzions Mango~        2
## # i 246,688 more rows
## # i 2 more variables: TOT_SALES <dbl>, pack_size <dbl>
```

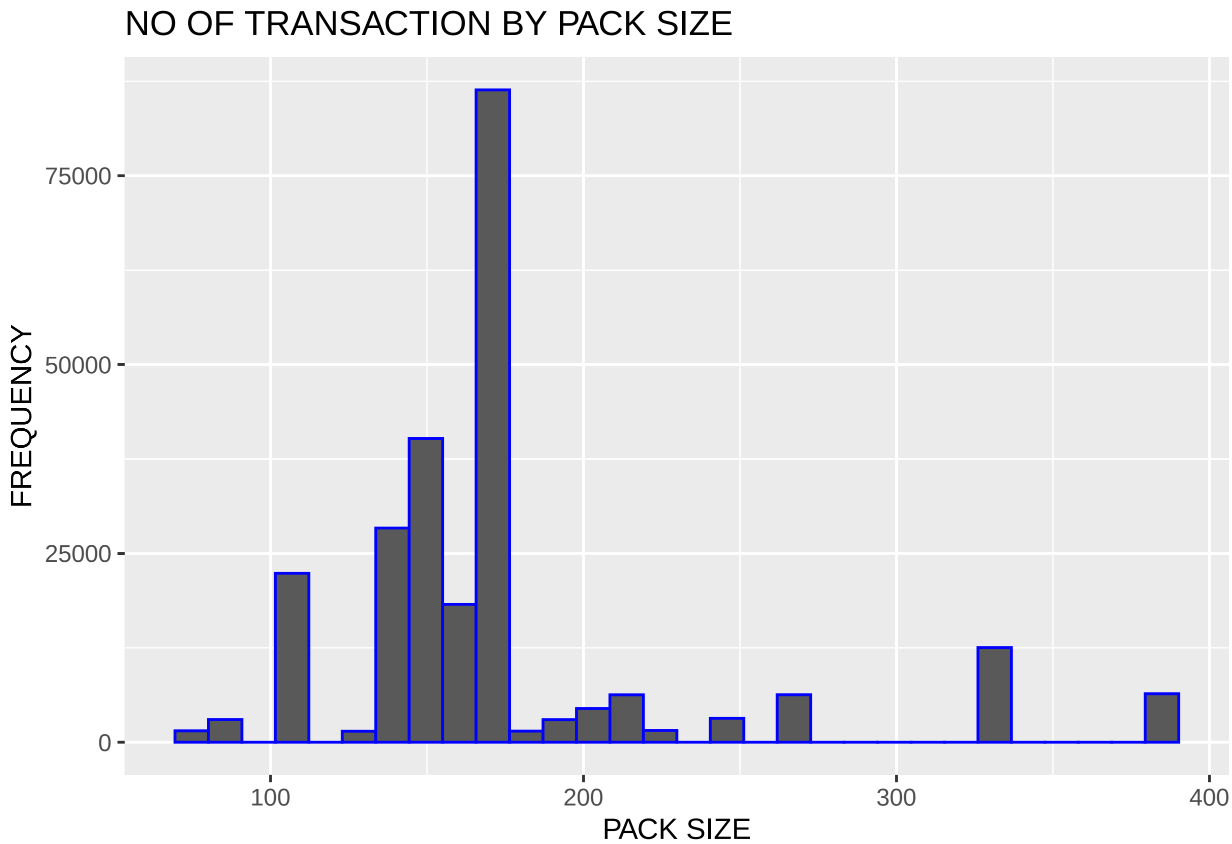
```
transaction_data %>%
  arrange(desc(pack_size))
```

```
## # A tibble: 246,698 x 9
##    DATE       STORE_NBR LYLTY_CARD_NBR TXN_ID PROD_NBR PROD_NAME        PROD_QTY
##    <date>         <dbl>          <dbl>  <dbl>    <dbl> <chr>               <dbl>
##  1 2019-05-20        55          55073  48887        4 Dorito Corn Chp~        1
##  2 2018-08-16        83          83186  83162        4 Dorito Corn Chp~        2
##  3 2019-05-14       130         130356 135147       14 Smiths Crnkle C~        2
##  4 2019-05-14       212         212203 211586        4 Dorito Corn Chp~        1
##  5 2019-05-19       257         257121 256483       14 Smiths Crnkle C~        1
##  6 2018-08-18       269         269175 266095        4 Dorito Corn Chp~        2
##  7 2018-11-03         3           3164   1779        4 Dorito Corn Chp~        2
##  8 2019-01-31         4           4072   2968       14 Smiths Crnkle C~        2
##  9 2018-12-12         4           4074   2980        4 Dorito Corn Chp~        2
## 10 2018-09-24         5           5050   4664        4 Dorito Corn Chp~        2
## # i 246,688 more rows
## # i 2 more variables: TOT_SALES <dbl>, pack_size <dbl>
```

Yeah, it makes sense, the minimum pack size is 70g and the highest is 380g. Now let's plot a histogram of Pack size, though it numeric but a categorical Variable.

```
ggplot(transaction_data, aes(pack_size)) +
  geom_histogram(col = "blue") +
  xlab("PACK SIZE")+ ylab("FREQUENCY") +
  ggtitle("NO OF TRANSACTION BY PACK SIZE")
```

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



Creating brand name colum

Now we create brand name column from PROD_NAME

```
transaction_data$brand <- word(transaction_data$PROD_NAME, 1)
transaction_data %>%
  group_by(brand) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

```
## # A tibble: 28 x 2
##    brand      count
##    <chr>      <int>
##  1 Kettle     41282
##  2 Smiths     27387
##  3 Pringles   25097
##  4 Doritos    22036
##  5 Thins      14074
##  6 RRD        11894
##  7 Infuzions  11054
##  8 WW         10320
##  9 Cobs        9688
## 10 Tostitos    9469
## # i 18 more rows
```

kettle is the most purchased brand follow by the smiths, however,Some brands are the same but with different names, now we clean the brand names.

```
transaction_data <- transaction_data %>%
  mutate(brand = recode(brand, RRD = "Red", Snbts = "Sunbites", Infzns ="Infuzions",WW ="Woolworths", Sh
## Re-examining the brands Variable
transaction_data %>%
  group_by(brand) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

```
## # A tibble: 20 x 2
##    brand      count
##    <chr>      <int>
##  1 Kettle     41282
##  2 Smiths     30350
##  3 Doritos    25218
##  4 Pringles   25097
##  5 Red        16321
##  6 Infuzions  14195
##  7 Thins      14074
##  8 Woolworths 11836
##  9 Cobs        9688
## 10 Tostitos    9469
## 11 Twisties    9453
## 12 Grnwves     7737
## 13 Natural     7469
## 14 Tyrrells    6439
## 15 Cheezels    4602
## 16 CCs         4551
## 17 Sunbites    3008
## 18 Cheetos     2927
## 19 Burger      1564
## 20 French      1418
```

Exploring Customer Data

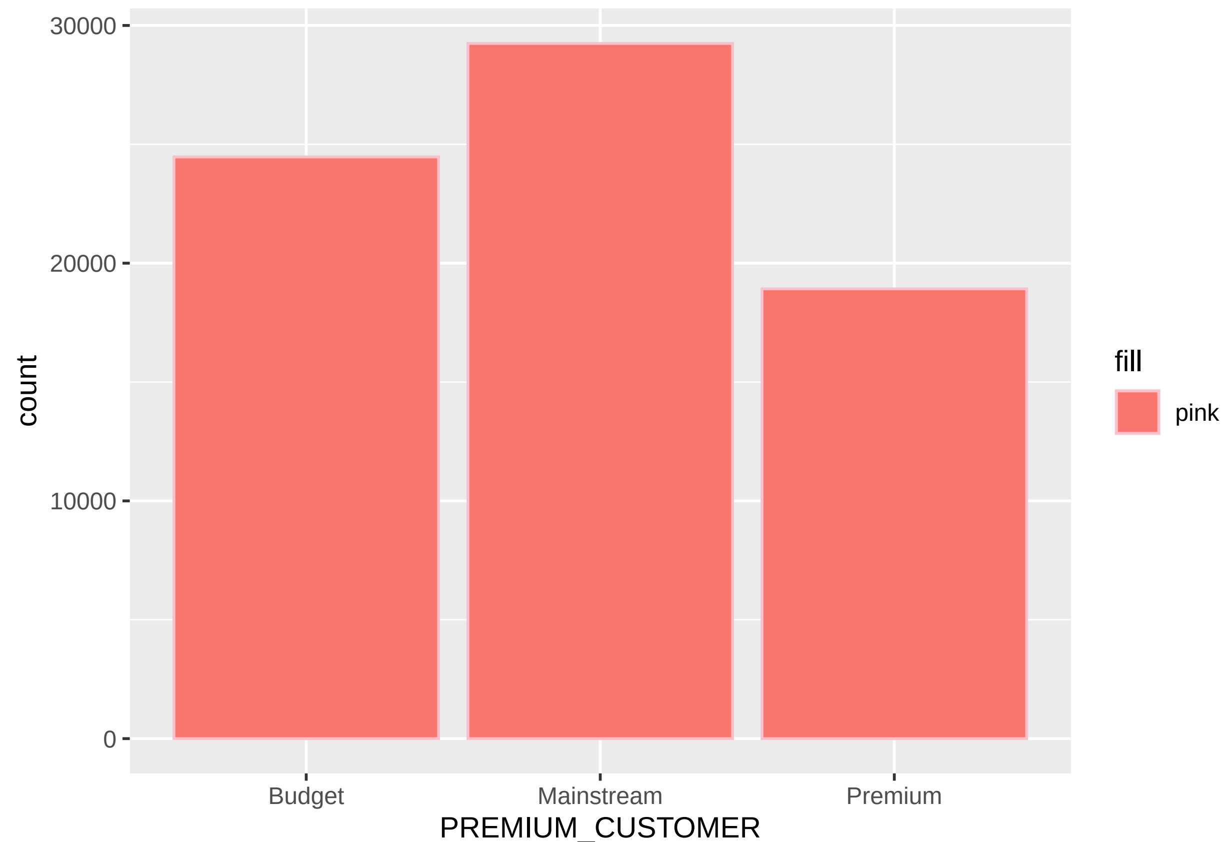
Now. we are satisfied with the transaction data, now we explore the customer data.

```
summary(purchase_behaviour)
```

```
##  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
```

```
## Distribution of Premium Customer
Premium_cust <- purchase_behaviour %>%
  group_by(PREMIUM_CUSTOMER) %>%
```

```r
  summarise(count = n())
ggplot(Premium_cust, aes(x = PREMIUM_CUSTOMER, y = count, fill = "pink")) +
  geom_col(col = "pink")
```



We can deduce the mainstream customers are more than any other group of customers. ## Joining the customer data to transaction data. Now we join the purchase behavior to the transaction table to find the transaction of all the customers.

```r
Data <- left_join(transaction_data, purchase_behaviour, by = "LYLTY_CARD_NBR")
```

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

```r
colSums(is.na(Data))
```

```
##             DATE        STORE_NBR   LYLTY_CARD_NBR           TXN_ID 
##                0                0                0                0 
##         PROD_NBR        PROD_NAME         PROD_QTY        TOT_SALES 
##                0                0                0                0 
##        pack_size            brand        LIFESTAGE PREMIUM_CUSTOMER 
##                0                0                0                0
```

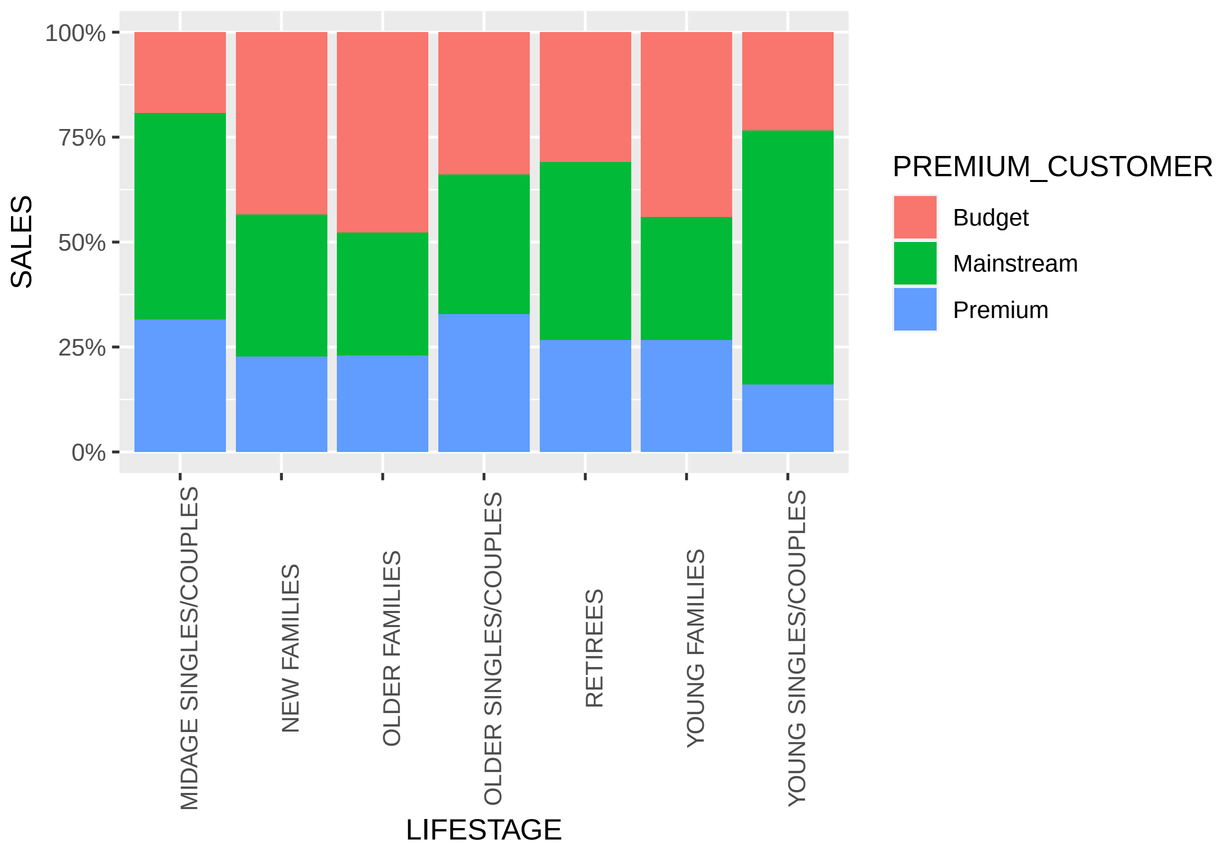
Here, we can see that no column has NA values, hence we can proceed with the analysis. ## DATA ANALYSIS ON CUSTOMER SEGMENTS Now that the data is ready we are ready to define some metrics of interest to the Client like: - Who spends the most on chips (total sales), describing customers by lifestage and how premium their general purchasing behaviour is - How many customers are in each segment - How many chips are bought per customer by segment - What's the average chip price by customer segment We could also ask our data team for more information. Examples are: - The customer's total spend over the period and total spend for each transaction to understand what proportion of their grocery spend is on chips - Proportion of customers in each customer segment overall to compare against the mix of customers who purchase chips ## Calculations We start with calculating the total sales by LIFESTAGE and PREMIUM_CUSTOMER an plotting the split by these segments to define which customer segment contribute to the most sales. ### Total Sales by LIFESTAGE AND PREMIUM CUSTOMER

```
sales <- Data %>%
  group_by(PREMIUM_CUSTOMER,LIFESTAGE) %>%
  summarise(SALES = sum(TOT_SALES))
```

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

Now we plot the 2 categorical variables against the continuous variable.

```
 ggplot(sales, aes(x = LIFESTAGE , y = SALES ,fill = PREMIUM_CUSTOMER, label = paste(SALES * 100, "%", s
 geom_bar(position = "fill", stat = "identity") +
  scale_y_continuous(labels = scales:: label_percent(accuracy = 1)) +
  theme(axis.text.x = element_text(angle = 90))
```



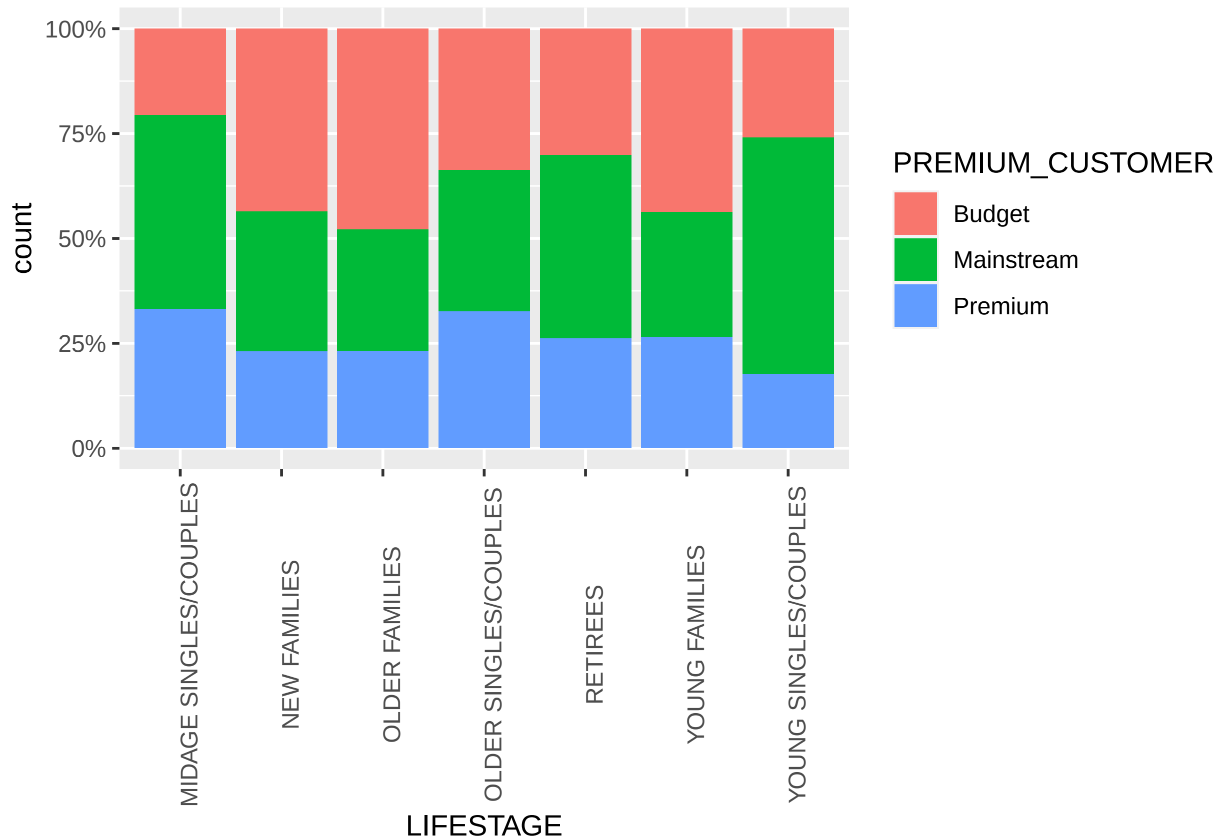
we can see that Sales are coming mainly from Budget - older families,Mainstream - young singles/couples, and Mainstream - retirees. ## Number of customers Now let's check if higher sales are also driven by the number of customers who buy chips.

```
no_of_cust <- Data %>%
  group_by(PREMIUM_CUSTOMER,LIFESTAGE) %>%
  summarise(count = n_distinct(LYLTY_CARD_NBR))
```

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

The plot showing the Proportion of sales

```
ggplot(no_of_cust, aes(x = LIFESTAGE , y = count ,fill = PREMIUM_CUSTOMER, label = paste(count * 100, "
 geom_bar(position = "fill", stat = "identity") +
  scale_y_continuous(labels = scales:: label_percent(accuracy = 1)) +
  theme(axis.text.x = element_text(angle = 90))
```



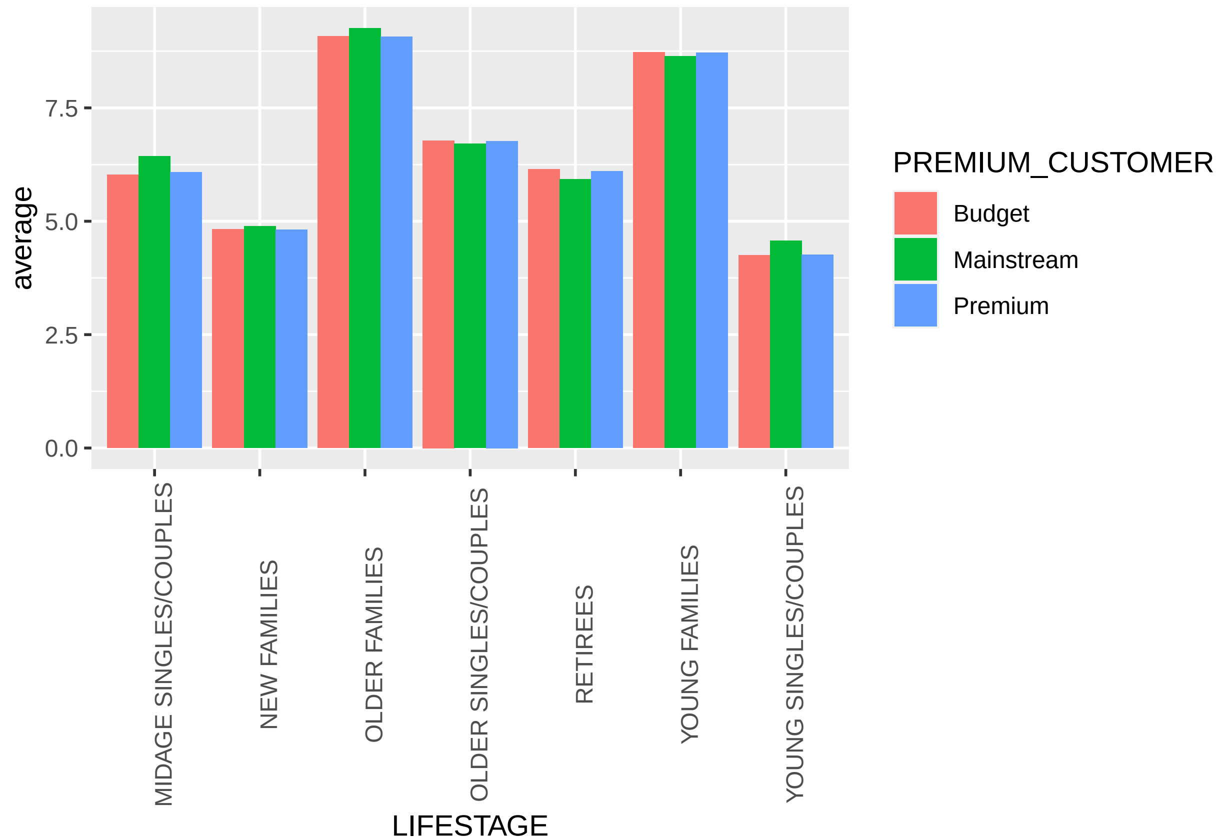
We can see that Mainstream-young/single couples and Mainstream- Retirees contribute to more sales but it is not a major driver for Budget - Older families segment. Higher sales may also be driven by more average units of chips being bought by each customer.

```
avg_purchase <- Data %>%
  group_by(PREMIUM_CUSTOMER,LIFESTAGE) %>%
  summarise(average = sum(PROD_QTY) / n_distinct(LYLTY_CARD_NBR))
```

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

Plotting the Average number of unit bought per customer by the 2 dimensions.

```
ggplot(avg_purchase, aes(x = LIFESTAGE , y = average ,fill = PREMIUM_CUSTOMER)) +
 geom_col(position=position_dodge()) +
  theme(axis.text.x = element_text(angle = 90))
```



We can see that in general, Older and young families buy more chips.

Average Price per unit chips bought by each customer segment

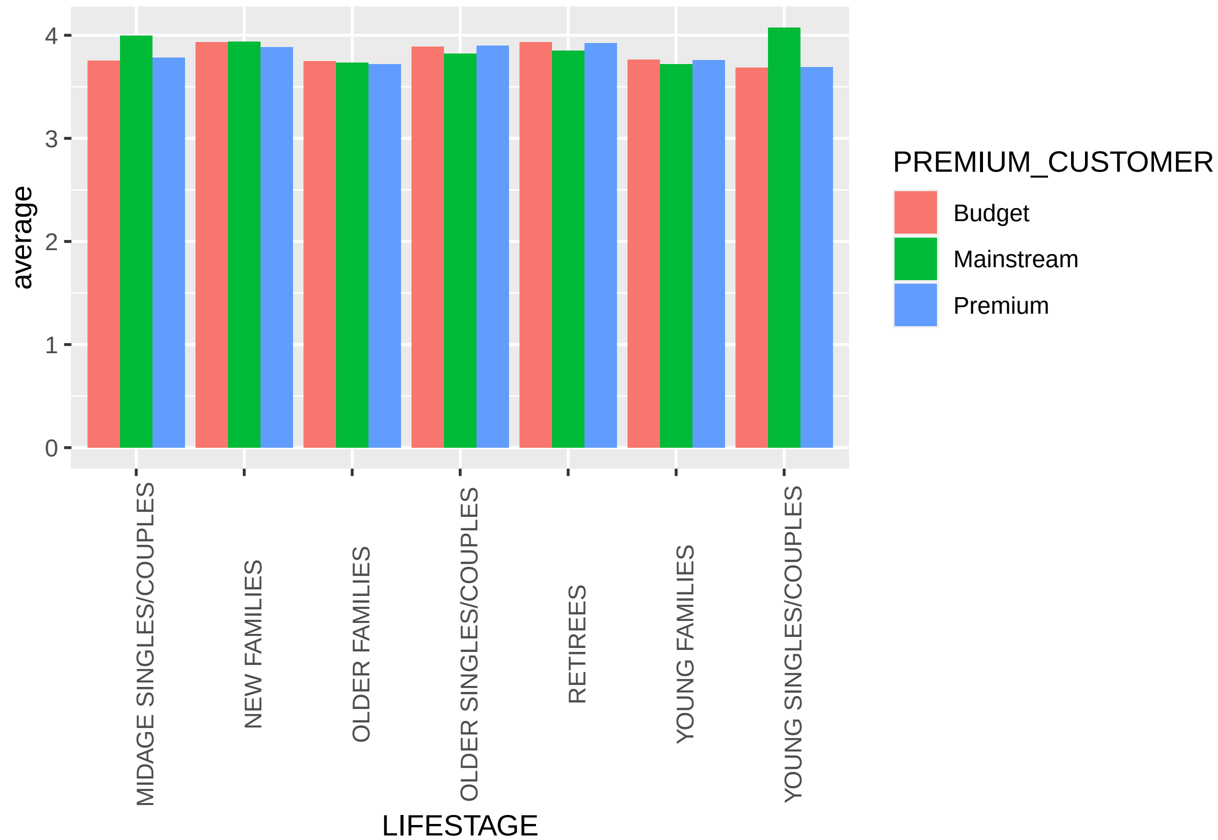
Let's also investigate the average price per unit chips bought for each customer segment as this is also a driver of total sales.

```
price_per_unit <- Data %>%
  group_by(PREMIUM_CUSTOMER,LIFESTAGE) %>%
  summarise(average =  sum(TOT_SALES) /sum(PROD_QTY))
```

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

Now, let's plot the average price per unit chips bought

```
ggplot(price_per_unit, aes(x = LIFESTAGE , y = average ,fill = PREMIUM_CUSTOMER)) +
 geom_col(position=position_dodge()) +
  theme(axis.text.x = element_text(angle = 90))
```



It is clear that Mainstream- Mid-age singles/couples are more willing to pay per packet of chips compared to Budget and premium counterparts. As the difference in price per unit isn't large, we can check for the statistical difference.

Perfoming Statistical Analysis

```
# Preparing a table for the analysis
PriceperUnit <- Data %>%
  mutate(price = TOT_SALES/PROD_QTY)
## Young and mid-age singles/couples that are Mainstream
PriceperUnit1 <- PriceperUnit %>%
  filter(LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES"), PREMIUM_CUSTOMER == "Mains
  select(LIFESTAGE,PREMIUM_CUSTOMER,price)
## Young and mid-age singles/couples that are not Mainstream.
PriceperUnit2 <- PriceperUnit %>%
  filter(LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES"), PREMIUM_CUSTOMER != "Mains
  select(LIFESTAGE,PREMIUM_CUSTOMER,price)
```

Perfoming T test

```
t.test(PriceperUnit1$price,PriceperUnit2$price, alternative = "greater")
```

```
##
##  Welch Two Sample t-test
##
## data:  PriceperUnit1$price and PriceperUnit2$price
## t = 37.612, df = 54791, p-value < 2.2e-16
## alternative hypothesis: true difference in means is greater than 0
## 95 percent confidence interval:
##  0.3186619       Inf
## sample estimates:
## mean of x mean of y
##  4.039726  3.706491
```

The t.test result in a p-value of 2.2e-16 i.e the unit price for mainstream, young and mid-age singles/couples ARE significantly higher than that of budget or premium, young and mid-age singles/ couples.

Deep dive into specific customer segments for insights

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

```
## Creating the target and other segment table
target_seg <- Data %>%
  filter(LIFESTAGE == "YOUNG SINGLES/COUPLES", PREMIUM_CUSTOMER == "Mainstream")
other <- Data %>%
  filter(LIFESTAGE != "YOUNG SINGLES/COUPLES", PREMIUM_CUSTOMER != "Mainstream")
```

Brand affirnity of target segment compared to others

```
target_seg <- target_seg  %>%
mutate(quantity_target_seg = sum(PROD_QTY))
other <- other  %>%
  mutate(quantity_other = sum(PROD_QTY))
### Quantity by Brand for the target segment
quantity_by_brand_target_seg <- target_seg %>%
  group_by(brand) %>%
  summarize(targetSegment = sum(PROD_QTY) / 36209)
##### Quantity by Brand for the others segment
quantity_by_brand_other <-other %>%
  group_by(brand) %>%
  summarize(otherSegment = sum(PROD_QTY) / 263506)

## Joining the resulting tables of groups
brand_proportions <- inner_join(quantity_by_brand_target_seg, quantity_by_brand_other, by  = "brand")
 ## calculating the brand affinity
```

```r
brand_proportions <- brand_proportions %>%
  mutate(affinityTobrand = targetSegment/otherSegment)
## Sorting in descending Order
brand_proportions %>%
  arrange(desc(affinityTobrand))
```

```
## # A tibble: 20 x 4
##    brand      targetSegment otherSegment affinityTobrand
##    <chr>              <dbl>        <dbl>           <dbl>
##  1 Tyrrells         0.0315       0.0257            1.23
##  2 Twisties         0.0462       0.0379            1.22
##  3 Doritos          0.123        0.101             1.21
##  4 Kettle           0.198        0.167             1.19
##  5 Tostitos         0.0453       0.0384            1.18
##  6 Pringles         0.119        0.101             1.18
##  7 Cobs             0.0447       0.0384            1.16
##  8 Infuzions        0.0645       0.0574            1.12
##  9 Thins            0.0604       0.0572            1.06
## 10 Grnwves          0.0327       0.0311            1.05
## 11 Cheezels         0.0180       0.0189            0.951
## 12 Smiths           0.0964       0.124             0.776
## 13 French           0.00395      0.00571           0.692
## 14 Cheetos          0.00804      0.0118            0.683
## 15 Red              0.0438       0.0672            0.652
## 16 Natural          0.0196       0.0310            0.633
## 17 CCs              0.0112       0.0184            0.606
## 18 Sunbites         0.00635      0.0126            0.504
## 19 Woolworths       0.0241       0.0488            0.495
## 20 Burger           0.00293      0.00654           0.448
```

Insight

We can see that, our target segment, Mainstream young/singles couples are 23% more likely to purchase Tyrells fore example compared to the others We can also see that our target segment - Mainstream young/singles couples are 55% less likely to Purchase Burger brand. ## Pack size purchase compared to others. Let's also find out if our target segment tends to buy larger packs of chips.

```r
## Quantity by pack size for target segment
quantity_by_pack_target_seg <- target_seg %>%
  group_by(pack_size) %>%
  summarize(targetSegment = sum(PROD_QTY) / 36209)
## Quantity by pack size for others
quantity_by_pack_other <- other %>%
  group_by(pack_size) %>%
  summarize(otherSegment = sum(PROD_QTY) / 263506)
## Joining the resulting tables of groups
pack_proportions <- inner_join(quantity_by_pack_target_seg, quantity_by_pack_other, by  = "pack_size")
 ## calculating the brand affinity
pack_proportions <- pack_proportions %>%
  mutate(affinityToPack = targetSegment/otherSegment)
pack_proportions %>%
  arrange(desc(affinityToPack))
```

```
## # A tibble: 20 x 4
##    pack_size targetSegment otherSegment affinityToPack
##        <dbl>         <dbl>        <dbl>          <dbl>
##  1       270       0.0318       0.0251           1.27
##  2       380       0.0322       0.0257           1.25
##  3       330       0.0613       0.0510           1.20
##  4       110       0.106        0.0896           1.19
##  5       134       0.119        0.101            1.18
##  6       210       0.0291       0.0249           1.17
##  7       135       0.0148       0.0129           1.14
##  8       250       0.0144       0.0129           1.12
##  9       170       0.0808       0.0804           1.01
## 10       150       0.158        0.163            0.967
## 11       175       0.255        0.271            0.939
## 12       165       0.0556       0.0616           0.903
## 13       190       0.00748      0.0121           0.617
## 14       180       0.00359      0.00618          0.581
## 15       160       0.00641      0.0122           0.524
## 16       125       0.00301      0.00598          0.504
## 17        90       0.00635      0.0126           0.504
## 18       200       0.00898      0.0185           0.486
## 19        70       0.00304      0.00628          0.483
## 20       220       0.00293      0.00654          0.448
```

we can see that mainstream - young/singles couples are 27% more likely to buy a 270g pack compared to the others. # Conclusion Let's recap our findings from the analysis, Sales have mainly been due to Budget - older families, Mainstream - young singles/couples, and mainstream retirees shoppers.We found the high spending coming from Mainstream young singles/couples and retirees is because they are more of them than any other buyers. We also found that Mainstream, mid-age, and young singles and couples are also more likely to pay more for chips.And also that mainstream - young singles/couples are 23% more likely to purchase Tyrells pack compared to other Segments.