Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021

Task 1

Data preparation and customer analytics

Conducting analysis on client's transaction dataset and identifying customer purchasing behaviours to generate insights and provide commercial recommendations.

Outline Of the main tasks to be looking for in the data for each.

Examine transaction data – look for inconsistencies, missing data across the data set, outliers, correctly identified category items, numeric data across all tables. If you determine any anomalies make the necessary changes in the dataset and save it. Having clean data will help when it comes to your analysis.

Examine customer data – check for similar issues in the customer data, look for nulls and when you are happy merge the transaction and customer data together so it’s ready for the analysis ensuring you save your files along the way.

Data analysis and customer segments – in your analysis make sure you define the metrics – look at total sales, drivers of sales, where the highest sales are coming from etc. Explore the data, create charts and graphs as well as noting any interesting trends and/or insights you find.

Deep dive into customer segments – define your recommendation from your insights, determine which segments we should be targeting, if packet sizes are relative and form an overall conclusion based on your analysis.

```
## Load required libraries and datasets
install.packages("data.table")
install.packages("ggmosaic")
library(data.table)
library(ggplot2)
library(readr)
library(readxl)
```

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021

Importing data files

QVI_purchase_behaviour <- read_csv("QVI_purchase_behaviour.csv")

QVI_transaction_data <- read_excel("QVI_transaction_data.xlsx")

##Viewing data files

View(QVI_purchase_behaviour)

View(QVI_transaction_data)

Exploratory data analysis

str(QVI_purchase_behaviour)

```
> str(QVI_purchase_behaviour)
spec_tbl_df [72,637 x 3] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ LYLTY_CARD_NBR  : num [1:72637] 1000 1002 1003 1004 1005 ...
 $ LIFESTAGE       : chr [1:72637] "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "OLDER SINGLES/COUPLES" ...
 $ PREMIUM_CUSTOMER: chr [1:72637] "Premium" "Mainstream" "Budget" "Mainstream" ...
 - attr(*, "spec")=
  .. cols(
  ..   LYLTY_CARD_NBR = col_double(),
  ..   LIFESTAGE = col_character(),
  ..   PREMIUM_CUSTOMER = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
```

str(QVI_transaction_data)

```
> str(QVI_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" "Smiths Crinkle Cut  Chips C
ken 170g" "Smiths Chip Thinly  S/Cream&Onion 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 ...
```

As we can see that the date column is in an integer format. So, we need to change this to a date format.

Examining PROD_NAME column

summary(QVI_transaction_data$PROD_NAME)

```
> summary(QVI_transaction_data$PROD_NAME)
   Length     Class      Mode 
   264836 character character 
```

##Examine the words in PROD_NAME to see if there are any incorrect entries #### such as products that are not chips

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

setnames(productWords, 'Products')
```

##Remove digits, and special characters, and then sort the distinct words by frequency of occurrence.

```
library(stringr)

library(stringi)
```

Removing digits

```
productWords$Products <- str_replace_all(productWords$Products,"[0-9]"," ")

productWords$Products <- str_replace_all(productWords$Products,"[gG]"," ")
```

Removing special characters

```
productWords$Products <- str_replace_all(productWords$Products,"[[:punct:]]"," ")
```

The most common words by counting the number of times a word appears

```
words <- strsplit(productWords$Products," ")

Products.freq<-table(unlist(words))
```

sorting them by this frequency in order of highest to lowest frequency

```
Products.freq <-  as.data.frame(Products.freq)

Products.freq <- Products.freq[order(Products.freq$Freq, decreasing = T),]
```

Remove salsa products

readdata <- function(fn){

QVI_transaction_data <- fread(fn)   ## no need to put a sep here, fread guess it

QVI_transaction_data[, SALSA := grepl("salsa", tolower(QVI_transaction_data$PROD_NAME))]

return(QVI_transaction_data)

QVI_transaction_data <- QVI_transaction_data[SALSA == FALSE,][, SALSA := NULL]

}

summary(QVI_transaction_data)

```
> summary(QVI_transaction_data)
      DATE              STORE_NBR     LYLTY_CARD_NBR        TXN_ID           PROD_NBR        PROD_NAME            PROD_QTY
 Min.   :2018-07-01   Min.   :  1.0   Min.   :   1000   Min.   :      1   Min.   :  1.00   Length:264836      Min.   :  1.000
 1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  70021   1st Qu.:  67602   1st Qu.: 28.00   Class :character   1st Qu.:  2.000
 Median :2018-12-30   Median :130.0   Median : 130358   Median : 135138   Median : 56.00   Mode  :character   Median :  2.000
 Mean   :2018-12-30   Mean   :135.1   Mean   : 135550   Mean   : 135158   Mean   : 56.58                      Mean   :  1.907
 3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202701   3rd Qu.: 85.00                      3rd Qu.:  2.000
 Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841   Max.   :114.00                      Max.   :200.000
   TOT_SALES
 Min.   :  1.500
 1st Qu.:  5.400
 Median :  7.400
 Mean   :  7.304
 3rd Qu.:  9.200
 Max.   :650.000
```

There are no nulls in the columns but product quantity appears to have an outlier which we should investigate further. Investigating further the case where 200 packets of chips are bought in one transaction.

install.packages("tidyverse")

library(tidyverse)

library(dplyr)

prod_qty_200 <- QVI_transaction_data %>% filter(PROD_QTY==200)

There are two transactions where 200 packets of chips are bought in one transaction and both of these transactions were by the same customer.

To see if the customer has had other transactions.

##Using a filter to see what other transactions that customer made

same_customer <- QVI_transaction_data %>% filter(LYLTY_CARD_NBR == 226000)

Filter out the customer based on the loyalty card number

QVI_transaction_data <- QVI_transaction_data[!(QVI_transaction_data$LYLTY_CARD_NBR == 226000),]

Re-examine transaction data

summary(QVI_transaction_data)

```
> summary(QVI_transaction_data)
      DATE               STORE_NBR     LYLTY_CARD_NBR        TXN_ID           PROD_NBR       PROD_NAME            PROD_QTY    
 Min.   :2018-07-01   Min.   :  1.0   Min.   :   1000   Min.   :      1   Min.   :  1.00   Length:264834      Min.   :1.000  
 1st Qu.:2018-09-30   1st Qu.: 70.0   1st Qu.:  70021   1st Qu.:  67601   1st Qu.: 28.00   Class :character   1st Qu.:2.000  
 Median :2018-12-30   Median :130.0   Median : 130357   Median : 135137   Median : 56.00   Mode  :character   Median :2.000  
 Mean   :2018-12-30   Mean   :135.1   Mean   : 135549   Mean   : 135158   Mean   : 56.58                      Mean   :1.906  
 3rd Qu.:2019-03-31   3rd Qu.:203.0   3rd Qu.: 203094   3rd Qu.: 202700   3rd Qu.: 85.00                      3rd Qu.:2.000  
 Max.   :2019-06-30   Max.   :272.0   Max.   :2373711   Max.   :2415841   Max.   :114.00                      Max.   :5.000  
   TOT_SALES     
 Min.   : 1.500  
 1st Qu.: 5.400  
 Median : 7.400  
 Mean   : 7.299  
 3rd Qu.: 9.200  
 Max.   :29.500  
```

Count the number of transactions by date

count_by_date <- count(QVI_transaction_data, QVI_transaction_data$DATE)

count_by_date

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021

```
> count_by_date <- count(QVI_transaction_data, QVI_transaction_data$DATE)
> count_by_date
# A tibble: 364 x 2
   `QVI_transaction_data$DATE`      n
   <date>                       <int>
 1 2018-07-01                     724
 2 2018-07-02                     711
 3 2018-07-03                     722
 4 2018-07-04                     714
 5 2018-07-05                     712
 6 2018-07-06                     762
 7 2018-07-07                     750
 8 2018-07-08                     696
 9 2018-07-09                     749
10 2018-07-10                     705
# ... with 354 more rows
```

##There are only 364 rows, meaning only 364 dates which indicates a missing date. Creating a sequence of dates from 1 Jul 2018 to 30 Jun 2019 and using this to create a chart of number of transactions over time to find the missing date.

Creating a sequence of dates and join this the count of transactions by date.

transaction_by_date <- QVI_transaction_data[order(QVI_transaction_data$DATE),]

Setting plot themes to format graphs

theme_set(theme_dark())

theme_update(plot.title = element_text(hjust = 0.5))

trans_Over_Time <-ggplot(count_by_date, aes(x = count_by_date$`QVI_transaction_data$DATE`, y = count_by_date$n)) +

geom_line() +

labs(x = "Day", y = "Number_of_transactions", title = "Transactions_over_time") +

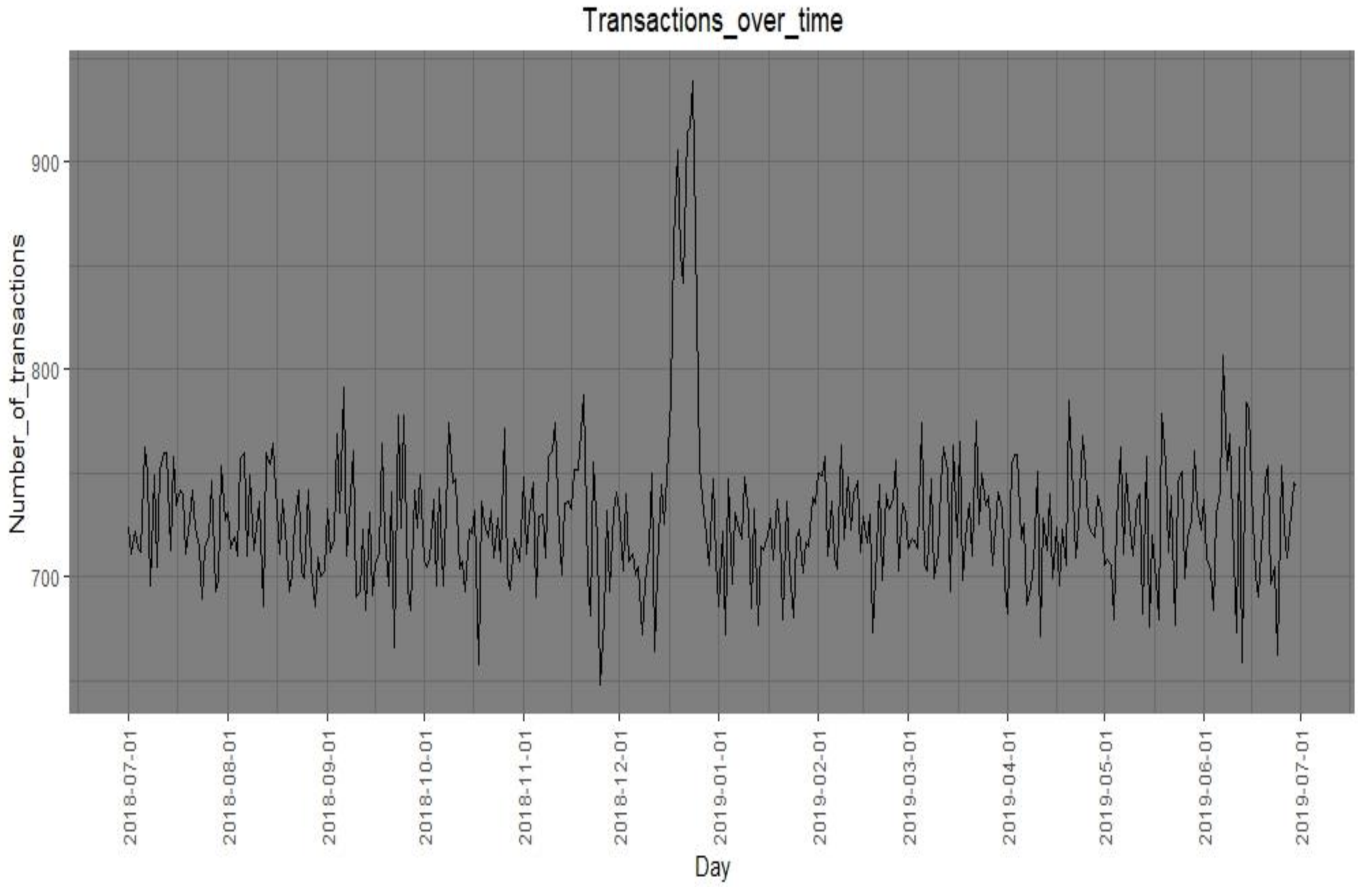
scale_x_date(breaks = "1 month") +

theme(axis.text.x = element_text(angle = 90, vjust = 0.5))

trans_Over_Time

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021



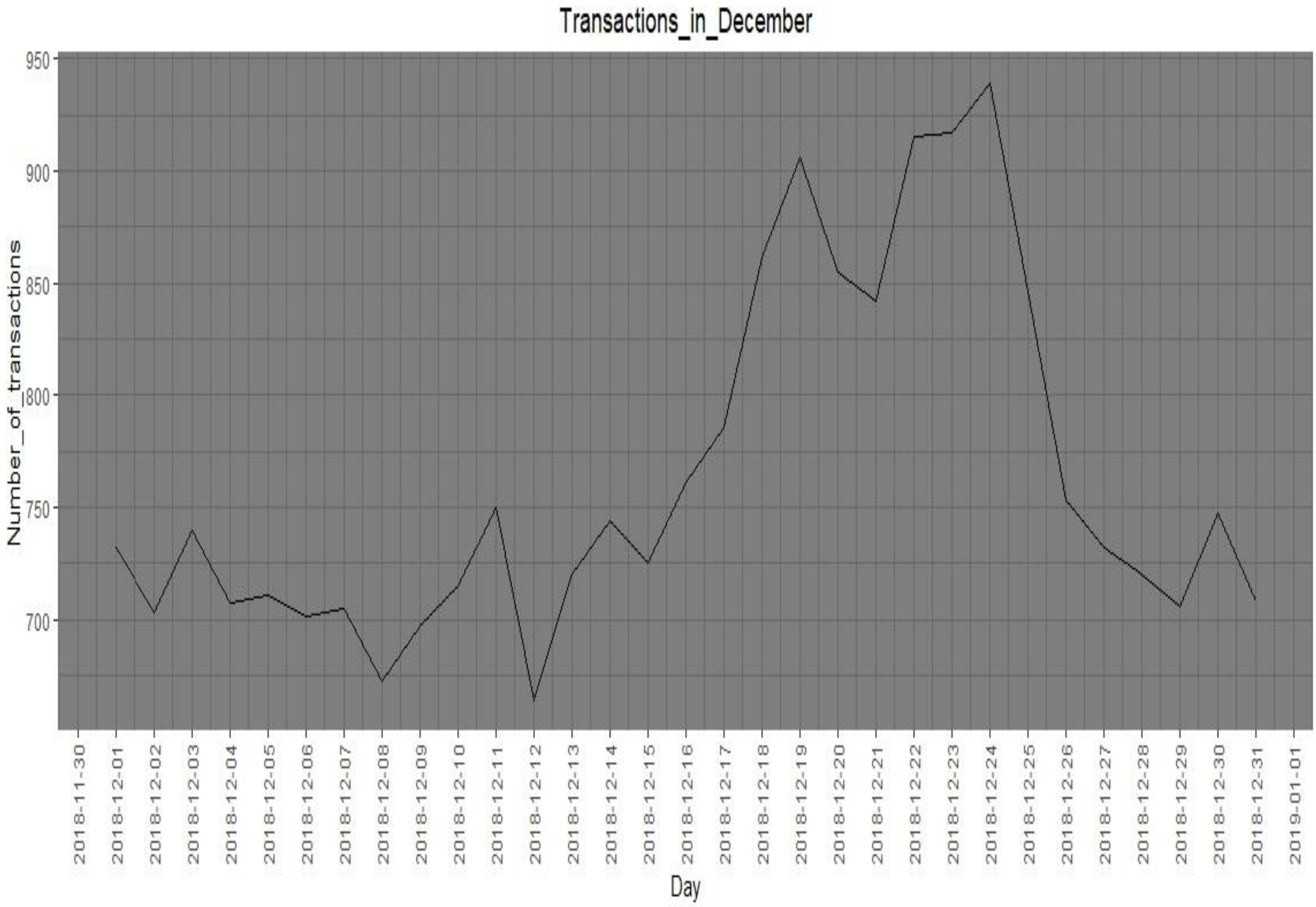
As its clear that there is an increase in purchases in December and a break in late December. Zooming in on this.

Filtering to December and looking at individual days.

```
filter_data <- count_by_date[
(count_by_date$`QVI_transaction_data$DATE` >= "2018-12-01" &
count_by_date$`QVI_transaction_data$DATE`<="2018-12-31"),]

ggplot(filter_data, aes(x =
filter_data$`QVI_transaction_data$DATE`, y = filter_data$n)) +

  geom_line() +

  labs(x = "Day", y = "Number_of_transactions", title =
"Transactions_in_December") +

  scale_x_date(breaks = "1 day") +

  theme(axis.text.x = element_text(angle = 90, vjust = 0.5))
```

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021



We can see that the increase in sales occurs in the lead-up to Christmas and that there are zero sales on Christmas day itself. This is due to shops being closed on Christmas day.

Now that we are satisfied that the data no longer has outliers, we can move on to creating other features such as brand of chips or pack size from PROD_NAME. We will start with pack size.

A new column PACK SIZE added to the data frame QVI_transaction_data

is.data.table(QVI_transaction_data)

data.table(QVI_transaction_data)

setDT(QVI_transaction_data)

QVI_transaction_data[, "PACK_SIZE" := parse_number(PROD_NAME)]

Always check your output

Check if the pack sizes look sensible

PackSize_Vs_Transactions <- QVI_transaction_data[, .N, PACK_SIZE][order(PACK_SIZE)]

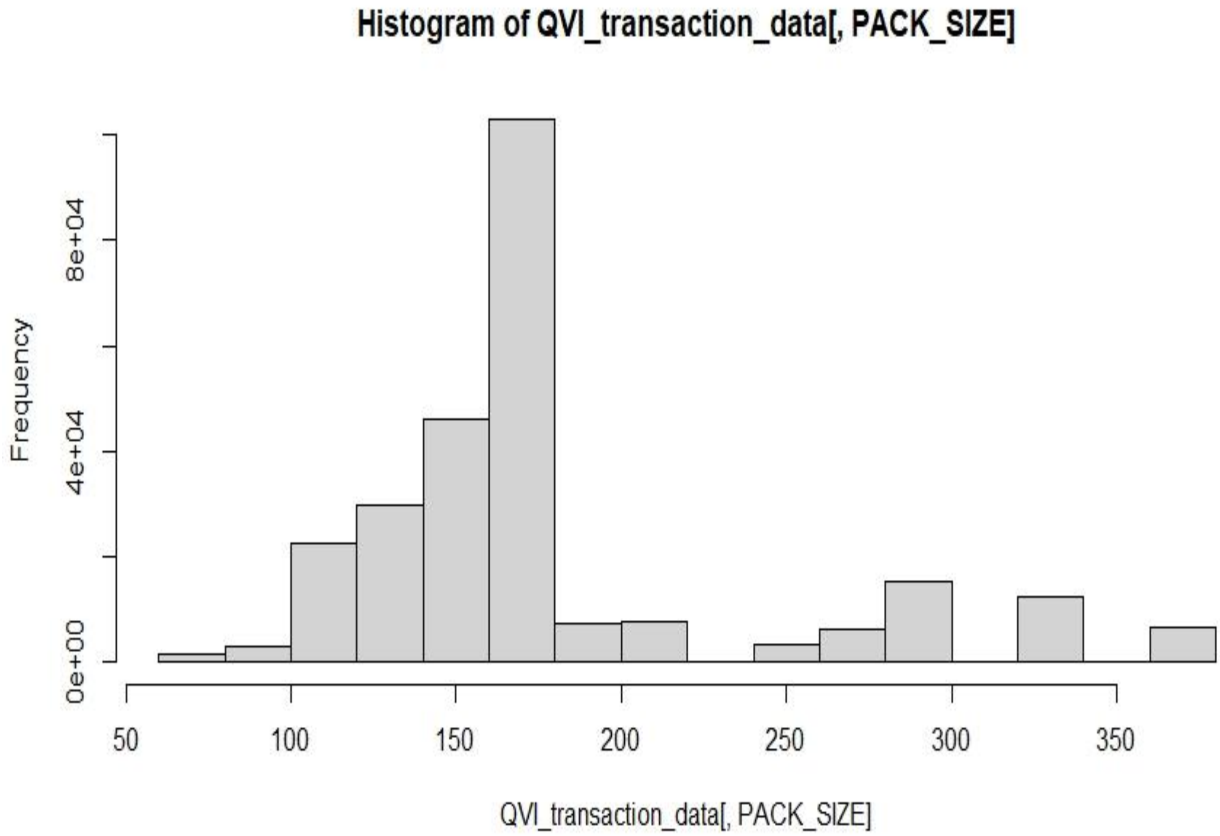
PackSize_Vs_Transactions

```
> PackSize_Vs_Transactions
    PACK_SIZE     N
 1:        70  1507
 2:        90  3008
 3:       110 22387
 4:       125  1454
 5:       134 25102
 6:       135  3257
 7:       150 43131
 8:       160  2970
 9:       165 15297
10:       170 19983
11:       175 66390
12:       180  1468
13:       190  2995
14:       200  4473
15:       210  6272
16:       220  1564
17:       250  3169
18:       270  6285
19:       300 15166
20:       330 12540
21:       380  6418
    PACK_SIZE     N
>
```

The largest size is 380g and the smallest size is 70g and that seems sensible!

Plotting a histogram of PACK_SIZE since we know that it is a categorical variable and not a continuous variable even though it is numeric.

A histogram showing the number of transactions by pack size.



Pack sizes created look reasonable.

##To create brands, we can use the first word in PROD_NAME to work out the brand name.

QVI_transaction_data$BRAND <- gsub("([A-Za-z]+).*", "\\1", QVI_transaction_data$PROD_NAME)

Checking brands

QVI_transaction_data[, .N, by = BRAND][order(-N)]

```
> QVI_transaction_data[, .N, by = BRAND][order(-N)]
         BRAND     N
 1:     Kettle 41288
 2:     Smiths 28860
 3:   Pringles 25102
 4:    Doritos 24962
 5:      Thins 14075
 6:        RRD 11894
 7:  Infuzions 11057
 8:         WW 10320
 9:       Cobs  9693
10:   Tostitos  9471
11:   Twisties  9454
12:        Old  9324
13:   Tyrrells  6442
14:      Grain  6272
15:    Natural  6050
16:        Red  5885
17:   Cheezels  4603
18:        CCs  4551
19: Woolworths  4437
20:     Dorito  3183
21:     Infzns  3144
22:      Smith  2963
23:    Cheetos  2927
24:      Snbts  1576
25:     Burger  1564
26:    GrnWves  1468
27:   Sunbites  1432
28:        NCC  1419
29:     French  1418
         BRAND     N
>
```

Some of the brand names look like they are of the same brands - such as RED and RRD, which are both Red Rock Deli chips. Combining these together.

Cleaning brand names

QVI_transaction_data[BRAND == "RED", BRAND := "RRD"]

QVI_transaction_data[BRAND == "SNBTS", BRAND := "SUNBITES"]

QVI_transaction_data[BRAND == "INFZNS", BRAND := "INFUZIONS"]

QVI_transaction_data[BRAND == "WW", BRAND := "WOOLWORTHS"]

QVI_transaction_data[BRAND == "SMITH", BRAND := "SMITHS"]

QVI_transaction_data[BRAND == "NCC", BRAND := "NATURAL"]

QVI_transaction_data[BRAND == "DORITO", BRAND := "DORITOS"]

QVI_transaction_data[BRAND == "GRAIN", BRAND := "GRNWVES"]

Checking again

QVI_transaction_data[, .N, by = BRAND][order(BRAND)]

```
> QVI_transaction_data[BRAND == "GRAIN", BRAND := "GRNWVES"]
> QVI_transaction_data[, .N, by = BRAND][order(BRAND)]
          BRAND     N
 1:      Burger  1564
 2:         CCs  4551
 3:     Cheetos  2927
 4:    Cheezels  4603
 5:        Cobs  9693
 6:      Dorito  3183
 7:     Doritos 24962
 8:      French  1418
 9:       Grain  6272
10:     GrnWves  1468
11:   Infuzions 11057
12:      Infzns  3144
13:      Kettle 41288
14:     NATURAL  1419
15:     Natural  6050
16:         Old  9324
17:    Pringles 25102
18:         RRD 11894
19:         Red  5885
20:       Smith  2963
21:      Smiths 28860
22:       Snbts  1576
23:    Sunbites  1432
24:       Thins 14075
25:    Tostitos  9471
26:    Twisties  9454
27:    Tyrrells  6442
28: WOOLWORTHS 10320
29: Woolworths  4437
          BRAND     N
>
```

Examining customer data. We are happy with the transaction dataset so looking at the customer dataset.

summary(QVI_purchase_behaviour)

```
> summary(QVI_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
>
```

Closer look at the LIFESTAGE and PREMIUM_CUSTOMER columns.

Merging transaction data to customer data

data1 <- merge(QVI_transaction_data, QVI_purchase_behaviour, all.x = TRUE)

As the number of rows in `data1` is the same as that of `QVI_transactionData`, we can be sure that no duplicates were created. This is because we created `data` by setting `all.x = TRUE` (in other words, a left join) which means take all the rows in `QVI_transactionData` and find rows with matching values in shared columns and then joining the details in these rows to the `x` or the first mentioned table.

Also checking if some customers were not matched on by checking for nulls.

apply(data1, 2, function(x) any(is.na(x)))

```
> apply(data1, 2, function(x) any(is.na(x)))
  LYLTY_CARD_NBR             DATE        STORE_NBR           TXN_ID         PROD_NBR        PROD_NAME         PROD_QTY
           FALSE            FALSE            FALSE            FALSE            FALSE            FALSE            FALSE
       TOT_SALES        PACK_SIZE            BRAND        LIFESTAGE PREMIUM_CUSTOMER
           FALSE            FALSE            FALSE            FALSE            FALSE
>
```

Since there are no nulls! So, all our customers in the transaction data has been accounted for in the customer dataset.

While continuing with Task 2, we will retain this dataset which we can write out as a csv.

write.csv(data1,"QVI_data1.csv")

Data exploration is now complete!

Data analysis on customer segments

Since the data is ready for analysis, we can define some metrics of interest to the client:

- Who spends the most on chips (total sales), describing customers by life stage 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

Calculating total sales by LIFESTAGE and PREMIUM_CUSTOMER

total_sales <- data1 %>% group_by(LIFESTAGE,PREMIUM_CUSTOMER)

pf.total_sales <- summarise(total_sales,sales_count=sum(TOT_SALES))

summary(pf.total_sales)

```
> summary(pf.total_sales)
  LIFESTAGE          PREMIUM_CUSTOMER    sales_count    
 Length:21          Length:21          Min.   : 11491  
 Class :character   Class :character   1st Qu.: 58433  
 Mode  :character   Mode  :character   Median : 92789  
                                       Mean   : 92053  
                                       3rd Qu.:133394  
                                       Max.   :168363  
```

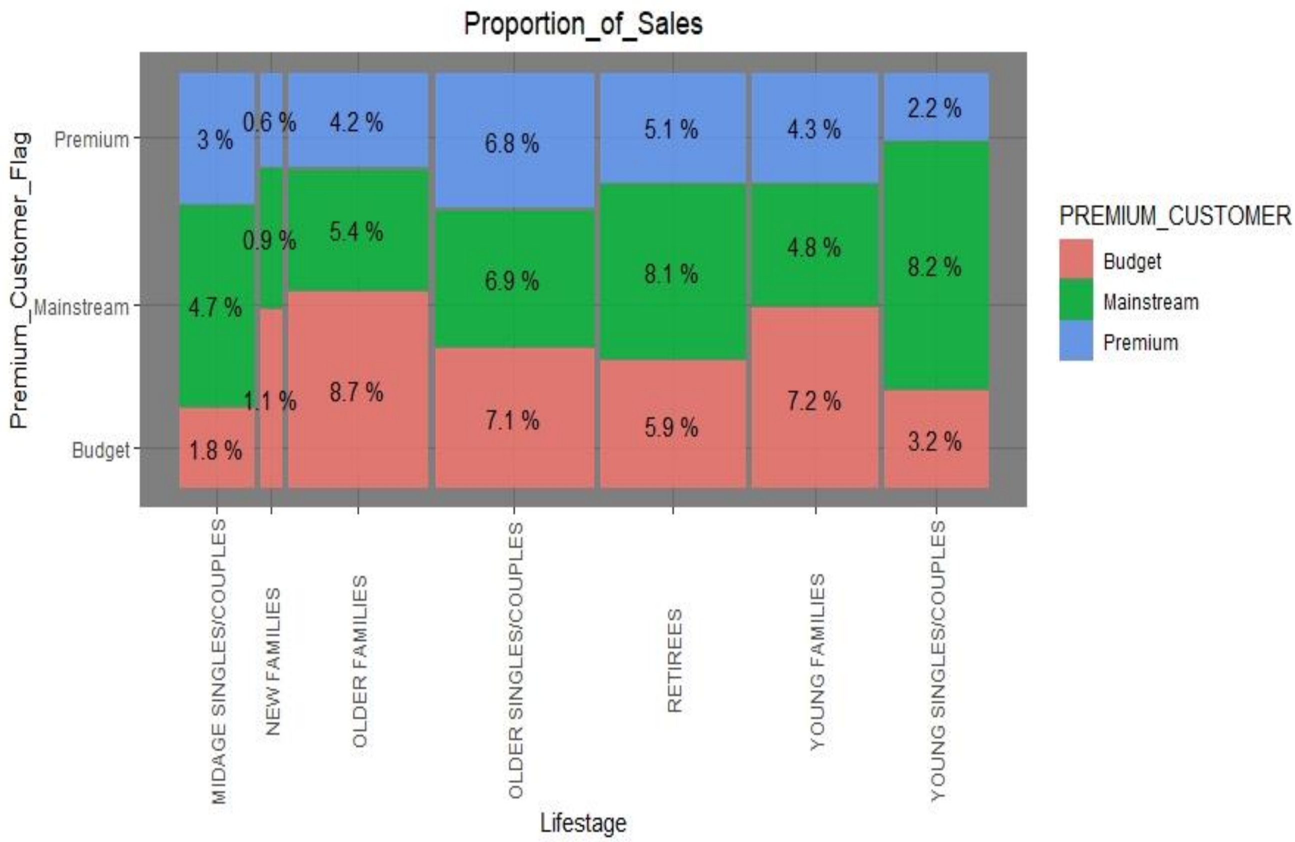
Plotting the split by these segments to describe which customer segment contribute most to chip sales.

p <- ggplot(pf.total_sales) + geom_mosaic(aes(weight = sales_count, x = product(PREMIUM_CUSTOMER, LIFESTAGE),fill = PREMIUM_CUSTOMER)) + labs(x = "Lifestage", y = "Premium_Customer_Flag", title = "Proportion_of_Sales") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5))

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, '%'))), inherit.aes = F)

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021



Sales are mainly from budget - older families, Mainstream - young singles/couples, and Mainstream – retirees.

Seeing if the higher sales are due to there being more customers who buy chips.

Number of customers by LIFESTAGE and PREMIUM_CUSTOMER

total_sales <- data1 %>% group_by(LIFESTAGE,PREMIUM_CUSTOMER)

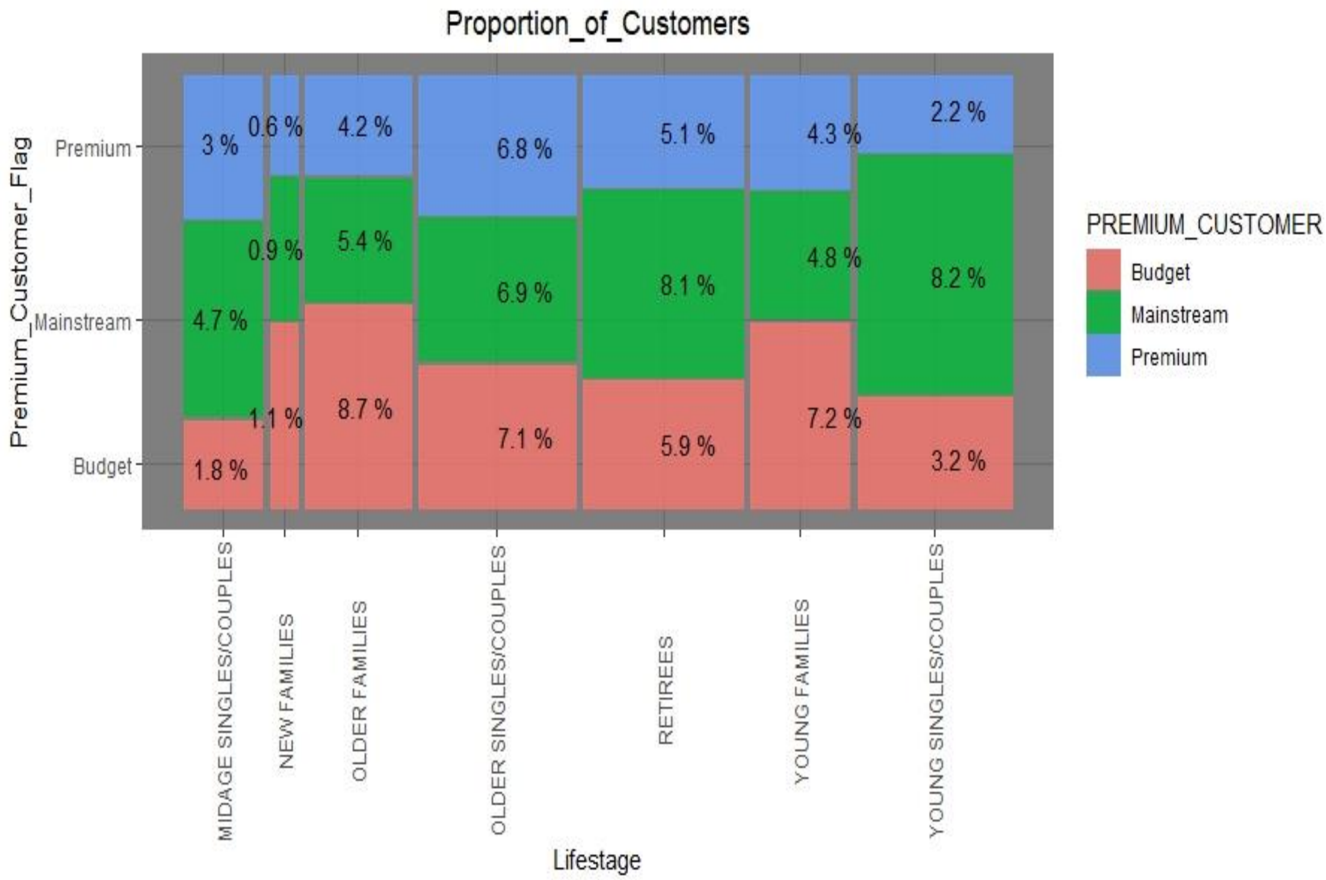
no_of_customers <- summarise(total_sales,customer_count = length(unique(LYLTY_CARD_NBR)))

summary(no_of_customers)

```
> summary(no_of_customers)
  LIFESTAGE          PREMIUM_CUSTOMER   customer_count
 Length:21          Length:21          Min.   : 588
 Class :character   Class :character   1st Qu.:2431
 Mode  :character   Mode  :character   Median :3340
                                       Mean   :3459
                                       3rd Qu.:4675
                                       Max.   :8088
>
```

creating the plot

```
pl <- ggplot(data = no_of_customers) + geom_mosaic(aes(weight = customer_count, x = product(PREMIUM_CUSTOMER, LIFESTAGE), fill = PREMIUM_CUSTOMER)) + labs(x = "Lifestage", y = "Premium_Customer_Flag", title = "Proportion_of_Customers") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5))+ 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, '%'))))

pl
```



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

Higher sales may also be driven by more units of chips being bought per customer. Having a look at this next.

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021

Average number of units per customer by LIFESTAGE and PREMIUM_CUSTOMER.

total_sales_1 <-data1 %>% group_by(LIFESTAGE,PREMIUM_CUSTOMER)

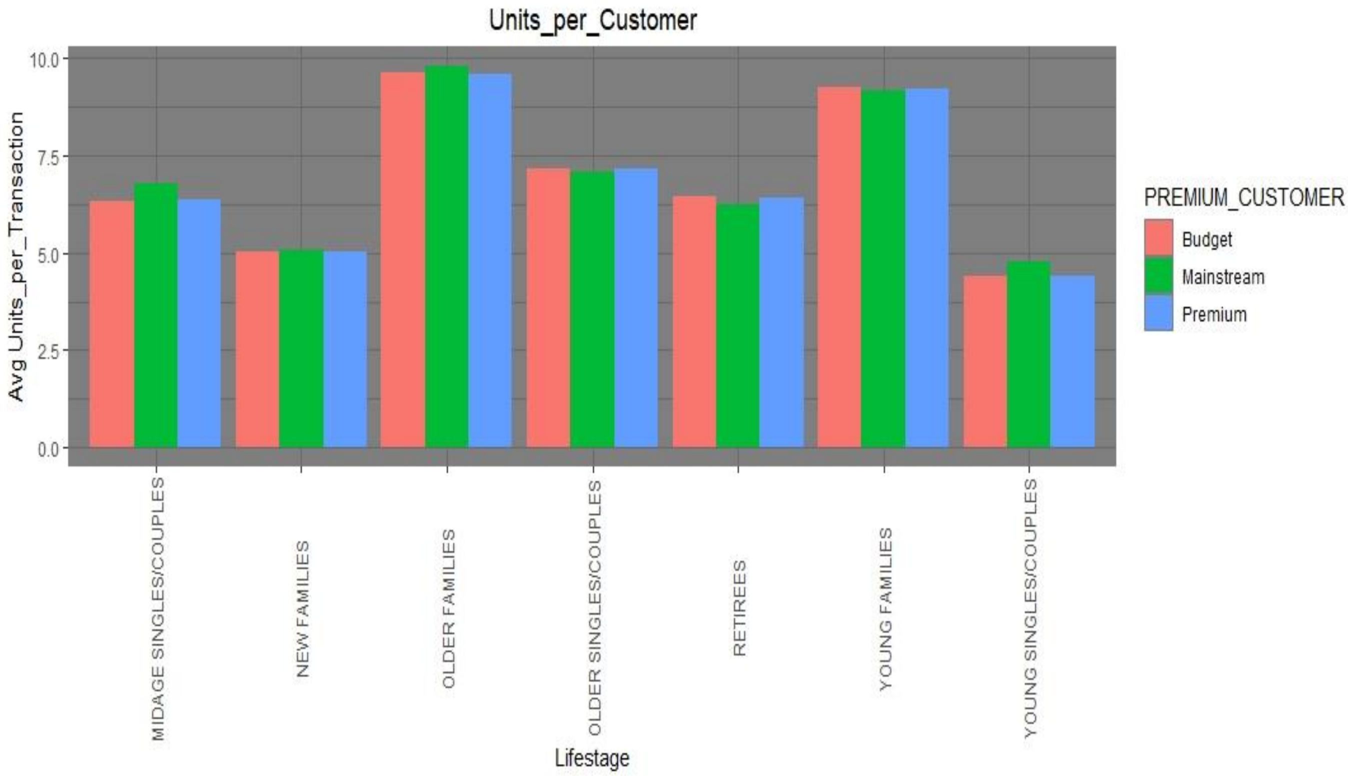
units <- summarise(total_sales_1, units_count = (sum(PROD_QTY)/uniqueN(LYLTY_CARD_NBR)))

summary(units)

Plotting the average number of units per customer by those two dimensions.

ggplot(data = units, aes(weight = units_count, x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) + geom_bar(position = position_dodge()) +

labs(x = "Lifestage", y = "Avg Units_per_Transaction", title = "Units_per_Customer") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5))



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

Also, Investigating the average price per unit chips bought for each customer segment as this is also a driver of total sales.

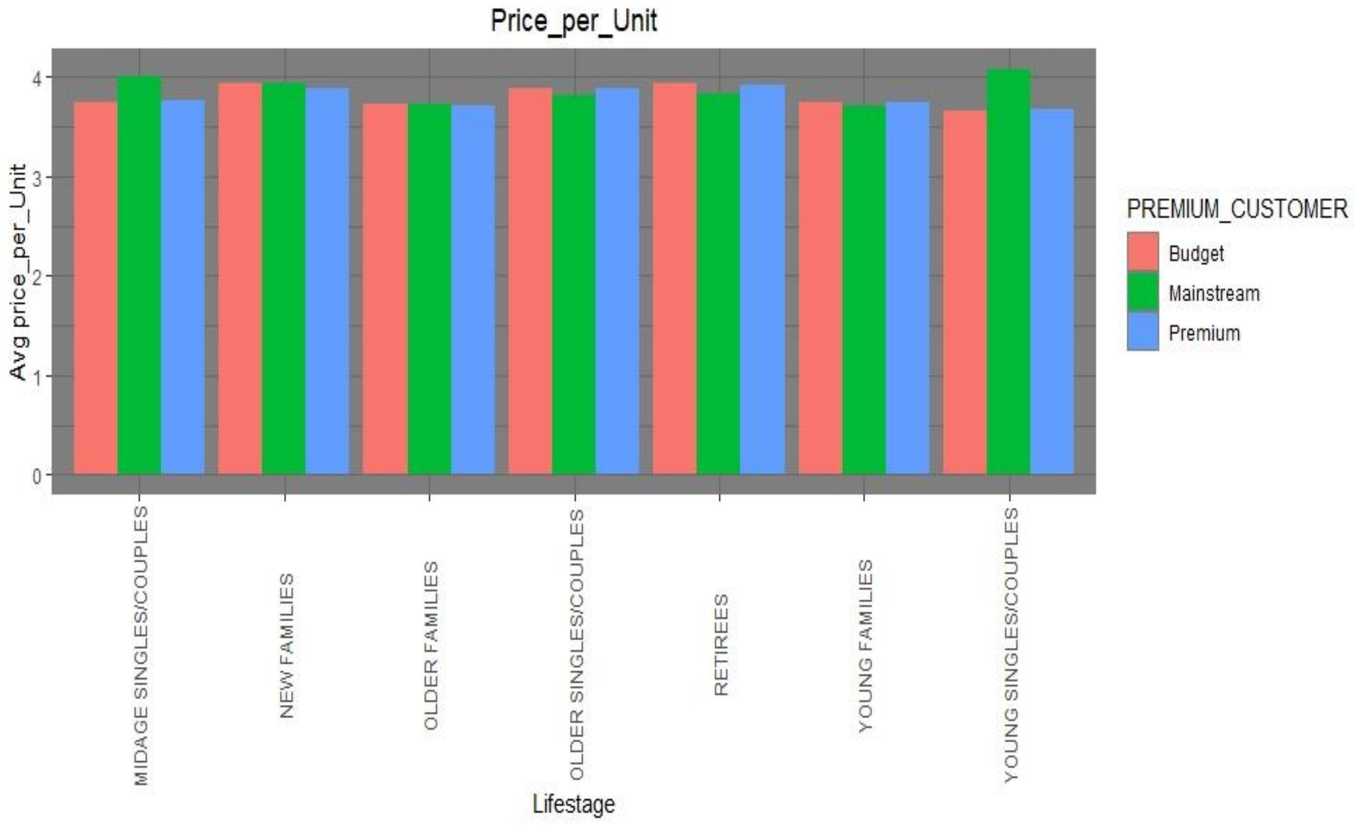
total_sales_2 <-data1 %>% group_by(LIFESTAGE,PREMIUM_CUSTOMER)

Average price per unit by LIFESTAGE and PREMIUM_CUSTOMER

PricePerUnit <- summarise(total_sales_2, price_per_unit = (sum(TOT_SALES)/sum(PROD_QTY)))

Plot the average price per unit sold (average sale price) by those two customer dimensions.

ggplot(data=PricePerUnit, aes(weight = price_per_unit,x = LIFESTAGE, fill = PREMIUM_CUSTOMER)) + geom_bar(position = position_dodge()) + labs(x = "Lifestage", y = "Avg price_per_Unit", title = "Price_per_Unit") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5))



Mainstream mid-Age and young singles and couples are more willing to pay more per packet of chips compared to their budget and premium counterparts. This may be due to premium shoppers being more likely to buy healthy snacks and when they buy chips, this is mainly for entertainment purposes rather than their own consumption.

This is also supported by there being fewer premium mid-Age and young singles and couples buying chips compared to their mainstream counterparts.

As the difference in average price per unit isn't large, we can check if this difference is statistically different.

Performing an independent t-test between mainstream vs premium and budget mid-Age and young singles and couples

PricePerUnit <- data1[, price := TOT_SALES/PROD_QTY]

t.test(data1[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER == "Mainstream", price],data1[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER != "Mainstream", price], alternative = "greater")

```
> t.test(data1[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER == "Mainstream", price],data1[LIFESTAGE %in% c("YO
UNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER != "Mainstream", price], alternative = "greater")

	Welch Two Sample t-test

data:  data1[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER == "Mainstream", price] and data1[LIFESTAGE %in% c
("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM_CUSTOMER != "Mainstream", price]
t = 40.61, df = 58792, p-value < 2.2e-16
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.3429435       Inf
sample estimates:
mean of x mean of y 
 4.045586  3.688165 
```

The t-test results in a p-value < 2.2e-16 i.e., the unit price for mainstream, young and mid-age singles and couples ARE SIGNIFICANTLY HIGHER than that of budget or premium, young and mid-Age singles and couples.

Deep dive into specific customer segments for insights.

We have found quite a few interesting insights that we can dive deeper into. We might want to target customer segments that contribute the most to sales to retain them or further increase sales.

Looking at Mainstream - young singles/couples. For instance, finding out if they tend to buy a particular brand of chips.

Diving deep into Mainstream, young singles/couples

segment1 <- data1[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER == "Mainstream",]

other <- data1[!(LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM_CUSTOMER =="Mainstream"),]

quantity_segment1 <- segment1[, sum(PROD_QTY)]

quantity_other <- other[, sum(PROD_QTY)]

quantity_segment1_by_brand <- segment1[, .(targetSegment = sum(PROD_QTY)/quantity_segment1), by = BRAND]

quantity_other_by_brand <- other[, .(other = sum(PROD_QTY)/quantity_other), by = BRAND]

brand_proportions <- merge(quantity_segment1_by_brand, quantity_other_by_brand)[, affinityToBrand := targetSegment/other]

brand_proportions[order(-affinityToBrand)]

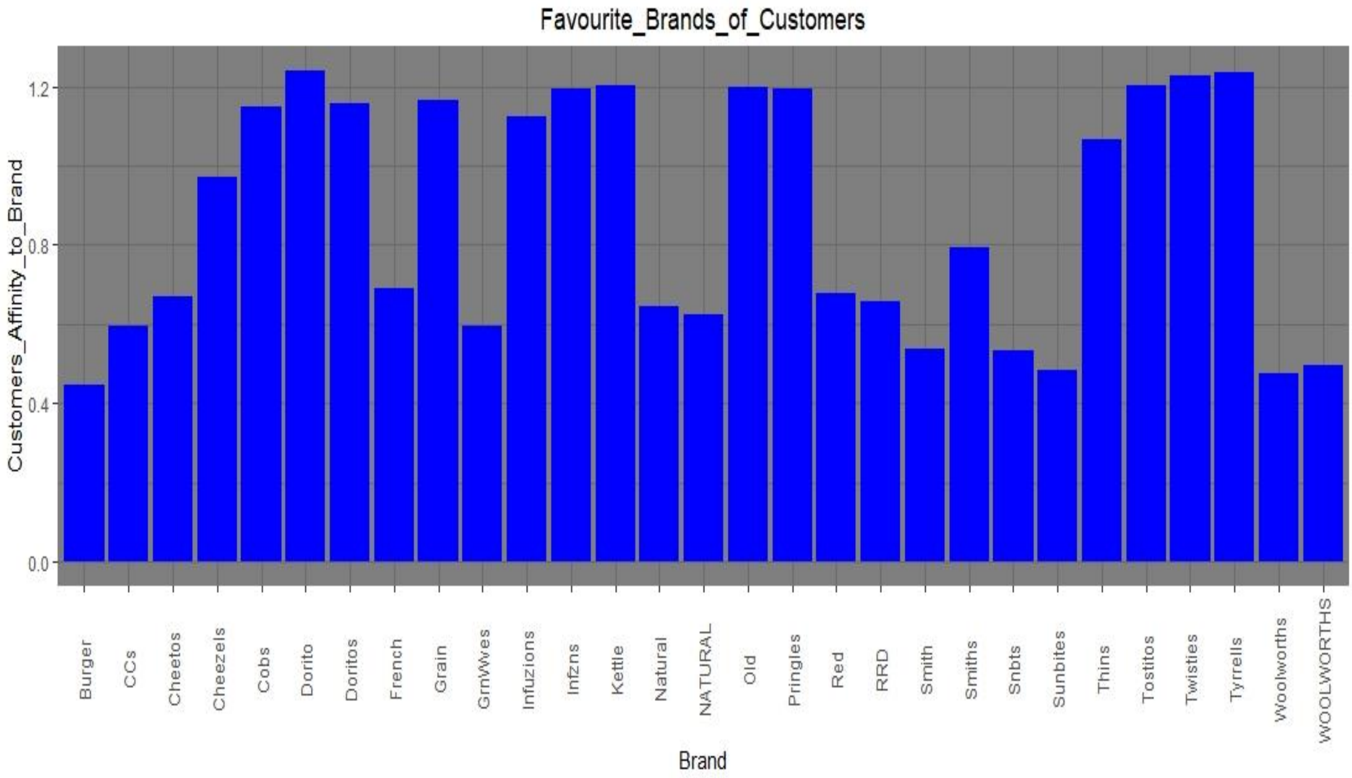
Quantium Data Analytics Virtual Experience Program Task-1


```
> brand_proportions[order(-affinityToBrand)]
         BRAND targetSegment       other affinityToBrand
 1:     Dorito   0.014728722 0.011886065       1.2391588
 2:   Tyrrells   0.029586871 0.023933043       1.2362352
 3:   Twisties   0.043306068 0.035282734       1.2274011
 4:     Kettle   0.185649203 0.154216335       1.2038232
 5:   Tostitos   0.042581280 0.035377136       1.2036384
 6:        Old   0.041597639 0.034752796       1.1969581
 7:     Infzns   0.014003935 0.011712280       1.1956626
 8:   Pringles   0.111979706 0.093743295       1.1945356
 9:      Grain   0.027308967 0.023400959       1.1670020
10:    Doritos   0.108148685 0.093391433       1.1580150
11:       Cobs   0.041856492 0.036374793       1.1507005
12:  Infuzions   0.046645268 0.041444608       1.1254846
13:      Thins   0.056611100 0.053083941       1.0664449
14:   Cheezels   0.016851315 0.017369961       0.9701412
15:     Smiths   0.087233382 0.110192837       0.7916429
16:     French   0.003701595 0.005363748       0.6901134
17:        Red   0.015349969 0.022641453       0.6779587
18:    Cheetos   0.007532615 0.011240270       0.6701454
19:        RRD   0.030026921 0.045784952       0.6558251
20:    Natural   0.014961690 0.023270084       0.6429581
21:    NATURAL   0.003416856 0.005471023       0.6245370
22:        CCs   0.010483537 0.017601675       0.5955988
23:    GrnWves   0.003365086 0.005651245       0.5954592
24:      Smith   0.006186581 0.011521331       0.5369676
25:      Snbts   0.003261545 0.006136128       0.5315315
26: WOOLWORTHS   0.019931663 0.040101525       0.4970300
27:   Sunbites   0.002692069 0.005582589       0.4822259
28: Woolworths   0.008257403 0.017327051       0.4765614
29:     Burger   0.002743839 0.006144710       0.4465369
         BRAND targetSegment       other affinityToBrand
```

ggplot(brand_proportions, aes(brand_proportions$BRAND,brand_proportions$affinityToBrand)) + geom_bar(stat = "identity",fill = "blue") + labs(x = "Brand", y = "Customers_Affinity_to_Brand", title = "Favourite_Brands_of_Customers") + theme(axis.text.x = element_text(angle = 90, vjust = 0.5))

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021



We can see that:

• 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

Also finding out if our target segment tends to buy larger packs of chips.

Preferred pack size compared to the rest of the population.

quantity_segment1_by_pack <- segment1[, .(targetSegment = sum(PROD_QTY)/quantity_segment1), by = PACK_SIZE]

quantity_other_by_pack <- other[, .(other = sum(PROD_QTY)/quantity_other), by = PACK_SIZE]

pack_proportions <- merge(quantity_segment1_by_pack, quantity_other_by_pack)[, affinityToPack := targetSegment/other]

pack_proportions[order(-affinityToPack)]

Quantium Data Analytics Virtual Experience Program Task-1

CHINNAM LAKSHMI DURGA 21-09-2021

```
> pack_proportions[order(-affinityToPack)]
    PACK_SIZE targetSegment       other affinityToPack
 1:       270   0.029845724 0.023377359      1.2766936
 2:       380   0.030156347 0.023832205      1.2653612
 3:       330   0.057465314 0.046726826      1.2298142
 4:       134   0.111979706 0.093743295      1.1945356
 5:       110   0.099658314 0.083642285      1.1914824
 6:       210   0.027308967 0.023400959      1.1670020
 7:       135   0.013848623 0.012179999      1.1369971
 8:       250   0.013460344 0.011905375      1.1306107
 9:       170   0.075740319 0.075440042      1.0039803
10:       300   0.054954442 0.057263373      0.9596787
11:       175   0.239102299 0.251516868      0.9506412
12:       150   0.155130462 0.163446272      0.9491221
13:       165   0.052184717 0.058003570      0.8996811
14:       190   0.007014910 0.011589987      0.6052561
15:       180   0.003365086 0.005651245      0.5954592
16:       160   0.006005384 0.011525622      0.5210464
17:        90   0.005953614 0.011718716      0.5080431
18:       125   0.002821495 0.005623353      0.5017460
19:       200   0.008412715 0.017378543      0.4840863
20:        70   0.002847380 0.005889395      0.4834759
21:       220   0.002743839 0.006144710      0.4465369
    PACK_SIZE targetSegment       other affinityToPack
```

We can see that the preferred PACK_SIZE is 270g.

data1[PACK_SIZE == 270, unique(PROD_NAME)]

```
> data1[PACK_SIZE == 270, unique(PROD_NAME)]
[1] "Twisties Cheese     270g" "Twisties Chicken270g"
```

CHINNAM LAKSHMI DURGA 21-09-2021

A FINAL INSIGHT:

- Sales have mainly been due to Budget - older families, Mainstream young singles/couples, and Mainstream - retirees shoppers.
- Found that the high spend in chips for mainstream young singles/couples and retirees is due to there being more of them than other buyers. Mainstream, Mid-Age and young singles and couples are also more likely to pay more per packet of chips. This is indicative of impulse buying behaviour.
- We've also found that Mainstream young singles and couples are 23% more likely to purchase Tyrrells chips compared to the rest of the population.
- The Category Manager may want to increase the category's performance by off-locating some Tyrrells and smaller packs of chips in discretionary space near segments where young singles and couples frequent more often to increase visibility and impulse behaviour.
- So, We can help the Category Manager with recommendations of where these segments are and further help them with measuring the impact of the changed placement.