# Task 1

# Quantium Virtual Internship

# Data preparation and customer analytics

# Retail Strategy and Analytics

# 

# Parisa Rahbarian

# 12 Oct 2021

Load required libraries:

**library(data.table)**

**library(ggplot2)**

**library(ggmosaic)**

**library(readr)**

Fill in the path to your working directory. If you are on a Windows machine, you will need to #use forward slashes (/) instead of backshashes (\)

**filePath <- "D:/Parisa/Internship/Quantium/"**

**transactionData <- fread(paste0(filePath,"QVI\_transaction\_data.csv"))**

**customerData <- fread(paste0(filePath,"QVI\_purchase\_behaviour.csv"))**

**Exploratory data analysis**

The first step in any analysis is to first understand the data. Let's take a look at each of the datasets provided.

**str(customerData)**

Classes ‘data.table’ and 'data.frame': 72637 obs. of 3 variables:

$ LYLTY\_CARD\_NBR : int 1000 1002 1003 1004 1005 1007 1009 1010 1011 1012 ...

$ LIFESTAGE : chr "YOUNG SINGLES/COUPLES" "YOUNG SINGLES/COUPLES" "YOUNG FAMILIES" "OLDER SINGLES/COUPLES" ...

$ PREMIUM\_CUSTOMER: chr "Premium" "Mainstream" "Budget" "Mainstream" ...

- attr(\*, ".internal.selfref")=<externalptr>

**str(transactionData)**

Classes ‘data.table’ and 'data.frame': 246740 obs. of 11 variables:

$ DATE : Date, format: "2018-10-17" "2019-05-14" ...

$ STORE\_NBR : int 1 1 1 2 2 4 4 5 7 7 ...

$ LYLTY\_CARD\_NBR: int 1000 1307 1343 2373 2426 4149 4196 5026 7150 7215 ...

$ TXN\_ID : int 1 348 383 974 1038 3333 3539 4525 6900 7176 ...

$ PROD\_NBR : int 5 66 61 69 108 16 24 42 52 16 ...

$ PROD\_NAME : chr "Natural Chip Compny SeaSalt175g" "CCs Nacho Cheese 175g" "Smiths Crinkle Cut Chips Chicken 170g" "Smiths Chip Thinly S/Cream&Onion 175g" ...

$ PROD\_QTY : int 2 3 2 5 3 1 1 1 2 1 ...

$ TOT\_SALES : num 6 6.3 2.9 15 13.8 5.7 3.6 3.9 7.2 5.7 ...

$ PACK\_SIZE : num 175 175 170 175 150 330 210 150 210 330 ...

$ BRAND : chr "Natural" "CCs" "Smiths" "Smiths" ...

$ PROD\_Brand : chr "Natural" "CCs" "Smiths" "Smiths" ...

- attr(\*, ".internal.selfref")=<externalptr>

- attr(\*, "index")= int(0)

..- attr(\*, "\_\_BRAND")= int [1:246740] 16 176 282 501 525 863 876 890 1146 1346 ..

**Examine transaction data**

**transactionData**

**head(transactionData)**

Convert DATE column to a date format #### A quick search online tells us that CSV and Excel #integer dates begin on 30 Dec 1899

We saw that the date format is in \_numeric\_ format which is wrong so we convert it to the `date` format as shown below\*\*

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

**Examine PROD\_NAME**

Changes! Looks like we are definitely looking at potato chips but how can we check that these are all chips? We can do some basic text analysis by summarising the individual words in the product name.

**summary(transactionData$PROD\_NAME)**

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(transactionData[, PROD\_NAME]), "**

**")))**

**setnames(productWords, 'Chips')**

**productWords**

#As we are only interested in words that will tell us if the product is chips or not, let's remove all #words with digits and special characters such as '&' from our set of product words. We can do this #using `grepl()`

**Removing digits**

**productWords$Chips <- str\_replace\_all(productWords$words,"[0-9]"," ")**

**productWords$Chips <- str\_replace\_all(productWords$words,"[gG]"," ")**

**Removing special characters**

Let's look at the most common words by counting the number of times a word appears and sorting them by this frequency in order of highest to lowest frequent

**productWords$Chips <- str\_replace\_all(productWords$Chips,"[[:punct:]]"," ")**

Let's look at the most common words by counting the number of times a word appears and

**wordsSep <- strsplit(productWords$Chips," ")**

**words.freq<-table(unlist(wordsSep))**

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

**words.freq <- as.data.frame(words.freq)**

**words.freq <- words.freq[order(words.freq$Freq, decreasing = T),]**

**words.freq**

Var1 Freq

1 234

15 175g 26

64 Chips 21

10 150g 19

2 & 17

176 Smiths 16

83 Crinkle 14

90 Cut 14

117 Kettle 13

50 Cheese 12

165 Salt 12

140 Original 10

61 Chip 9

96 Doritos 9

164 Salsa 9

8 134g 8

13 165g 8

14 170g 8

79 Corn 8

154 Pringles 8

161 RRD 8

6 110g 7

53 Chicken 7

221 WW 7

24 300g 6

168 Sea 6

180 Sour 6

57 Chilli 5

85 Crisps 5

203 Thinly 5

204 Thins 5

214 Vinegar 5

25 330g 4

81 Cream 4

93 Deli 4

113 Infuzions 4

133 Natural 4

156 Red 4

160 Rock 4

194 Supreme 4

18 200g 3

46 CCs 3

76 Cobs 3

94 Dip 3

97 El 3

120 Lime 3

127 Mild 3

136 Old 3

143 Paso 3

148 Popd 3

171 Sensations 3

184 Soy 3

195 Sweet 3

207 Tomato 3

208 Tortilla 3

209 Tostitos 3

211 Twisties 3

220 Woolworths 3

12 160g 2

17 190g 2

26 380g 2

28 90g 2

30 And 2

37 BBQ 2

44 Burger 2

51 Cheetos 2

52 Cheezels 2

62 ChipCo 2

65 Chives 2

101 French 2

106 Grain 2

108 Honey 2

119 Lightly 2

124 Medium 2

132 Nacho 2

151 Potato 2

158 Rings 2

167 Salted 2

175 Smith 2

188 SR 2

198 Swt 2

200 Tangy 2

202 Thai 2

212 Tyrrells 2

217 Waves 2

3 &Chives 1

4 &OnionStacked 1

5 &Sr/Cream 1

7 125g 1

9 135g 1

11 150G 1

16 180g 1

19 210g 1

20 210G 1

21 220g 1

22 250g 1

23 270g 1

27 70g 1

29 Aioli 1

31 Bacon 1

32 Bag 1

33 Balls 1

34 Barbecue 1

35 Barbeque 1

36 Basil 1

38 BBQ&Maple 1

39 Belly 1

40 Big 1

41 Bolognese 1

42 Box 1

43 Btroot 1

45 Camembert 1

47 Chckn175g 1

48 Ched 1

49 Cheddr&Mstrd 1

54 Chicken270g 1

55 Chikn&Garlic 1

56 Chili 1

58 Chilli& 1

59 Chilli&Lime 1

60 Chimuchurri 1

63 Chipotle 1

66 Chli&S/Cream175G 1

67 Chnky 1

68 Chp 1

69 ChpsBtroot&Ricotta 1

70 ChpsFeta&Garlic 1

71 ChpsHny&Jlpno 1

72 Chs 1

73 Chs&Onion170g 1

74 Chutny 1

75 Co 1

77 Coconut 1

78 Compny 1

80 Crackers 1

82 Cream&Chives 1

84 Crips 1

86 Crm 1

87 Crn 1

88 Crnchers 1

89 Crnkle 1

91 CutSalt/Vinegr175g 1

92 D/Style 1

95 Dorito 1

98 Fig 1

99 Flavour 1

100 Frch/Onin 1

102 FriedChicken 1

103 Fries 1

104 Garden 1

105 Gcamole 1

107 GrnWves 1

109 Hony 1

110 Hot 1

111 Hrb&Spce 1

112 Ht300g 1

114 Infzns 1

115 Jalapeno 1

116 Jam 1

118 Light& 1

121 Mac 1

122 Mango 1

123 Med 1

125 Mexican 1

126 Mexicana 1

128 Mozzarella 1

129 Mystery 1

130 Mzzrlla 1

131 N 1

134 NCC 1

135 Of 1

137 Onion 1

138 OnionDip 1

139 Orgnl 1

141 Originl 1

142 Papadums 1

144 Pc 1

145 Pepper 1

146 Pesto 1

147 Plus 1

149 Pork 1

150 Pot 1

152 PotatoMix 1

153 Prawn 1

155 Puffs 1

157 Rib 1

159 Roast 1

162 Rst 1

163 S/Cream&Onion 1

166 saltd 1

169 SeaSalt175g 1

170 Seasonedchicken 1

172 Siracha 1

173 Slow 1

174 Slt 1

177 Smoked 1

178 Snag&Sauce 1

179 Snbts 1

181 SourCream 1

182 SourCream&Herbs 1

183 Southern 1

185 Sp 1

186 Spicy 1

187 Splash 1

189 Stacked 1

190 Steak 1

191 Sthrn 1

192 Strws 1

193 Sunbites 1

196 Sweet&Spcy 1

197 SweetChili 1

199 Swt/Chlli 1

201 Tasty 1

205 Tmato 1

206 Tom 1

210 Truffle 1

213 Veg 1

215 Vinegr 1

216 Vingar 1

218 Whlegrn 1

219 Whlgrn 1

Looks like that the second most frequent word is chips.

**transactionData[!grepl("&",productWords),]**

**tail(names(sort(table(transactionData$PROD\_NAME))), 1)**

**Remove salsa products**

transactionData[, SALSA := grepl("salsa", tolower(PROD\_NAME))]

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

summary(transactionData)

DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR

Min. :2018-07-01 Min. : 1.0 Min. : 1000 Min. : 1 Min. : 1.00

1st Qu.:2018-09-30 1st Qu.: 70.0 1st Qu.: 70015 1st Qu.: 67569 1st Qu.: 26.00

Median :2018-12-30 Median :130.0 Median : 130367 Median : 135182 Median : 53.00

Mean :2018-12-30 Mean :135.1 Mean : 135530 Mean : 135130 Mean : 56.35

3rd Qu.:2019-03-31 3rd Qu.:203.0 3rd Qu.: 203083 3rd Qu.: 202652 3rd Qu.: 87.00

Max. :2019-06-30 Max. :272.0 Max. :2373711 Max. :2415841 Max. :114.00

PROD\_NAME PROD\_QTY TOT\_SALES PACK\_SIZE BRAND

Length:246740 Min. :1.000 Min. : 1.700 Min. : 70.0 Length:246740

Class :character 1st Qu.:2.000 1st Qu.: 5.800 1st Qu.:150.0 Class :character

Mode :character Median :2.000 Median : 7.400 Median :170.0 Mode :character

Mean :1.906 Mean : 7.316 Mean :175.6

3rd Qu.:2.000 3rd Qu.: 8.800 3rd Qu.:175.0

Max. :5.000 Max. :29.500 Max. :380.0

PROD\_Brand

Length:246740

Class :character

Mode :character

Summarise the data to check for nulls and possible outliers

**is.null(transactionData$PROD\_NAME)**

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

**Filter the dataset to find the outlier**

**library(tidyverse)**

**library(dplyr)**

**prod\_qty\_200 <- transactionData %>% filter(PROD\_QTY==200)**

DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR PROD\_NAME

1: 2018-08-19 226 226000 226201 4 Dorito Corn Chp Supreme 380g

2: 2019-05-20 226 226000 226210 4 Dorito Corn Chp Supreme 380g

PROD\_QTY TOT\_SALES

1: 200 650

2: 200 650

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

Let's see if the customer has had other transactions se I used a filter to see what other transactions that customer made.

**same\_customer <- transactionData %>% filter(LYLTY\_CARD\_NBR == 226000)**

It looks like this customer has only had the two transactions over the year and is not an ordinary retail customer. The customer might be buying chips for commercial purposes instead. I removed this loyalty card number from further analysis.

**transactionData <- transactionData[!(transactionData$LYLTY\_CARD\_NBR == 226000)]**

**Re-examine transaction data**

**summary(transactionData)**

DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR

Min. :2018-07-01 Min. : 1.0 Min. : 1000 Min. : 1 Min. : 1.00

1st Qu.:2018-09-30 1st Qu.: 70.0 1st Qu.: 70015 1st Qu.: 67569 1st Qu.: 26.00

Median :2018-12-30 Median :130.0 Median : 130367 Median : 135182 Median : 53.00

Mean :2018-12-30 Mean :135.1 Mean : 135530 Mean : 135130 Mean : 56.35

3rd Qu.:2019-03-31 3rd Qu.:203.0 3rd Qu.: 203083 3rd Qu.: 202652 3rd Qu.: 87.00

Max. :2019-06-30 Max. :272.0 Max. :2373711 Max. :2415841 Max. :114.00

PROD\_NAME PROD\_QTY TOT\_SALES

Length:246740 Min. :1.000 Min. : 1.700

Class :character 1st Qu.:2.000 1st Qu.: 5.800

Mode :character Median :2.000 Median : 7.400

Mean :1.906 Mean : 7.316

3rd Qu.:2.000 3rd Qu.: 8.800

Max. :5.000 Max. :29.500

That's better. Now, let's look at the number of transaction lines over time to see if there are any obvious data issues such as missing data.

**Count the number of transactions by date**

**countByDate <- count(transactionData, transactionData$DATE)**

**countByDate**

**transactionData$DATE n**

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

---

360: 2019-06-26 657

361: 2019-06-27 669

362: 2019-06-28 673

363: 2019-06-29 703

364: 2019-06-30 704

**nrow(countByDate)**

[1] 364

**summary(countByDate)**

transactionData$DATE n

Min. :2018-07-01 Min. :607.0

1st Qu.:2018-09-29 1st Qu.:658.0

Median :2018-12-30 Median :674.0

Mean :2018-12-30 Mean :677.9

3rd Qu.:2019-03-31 3rd Qu.:694.2

Max. :2019-06-30 Max. :865.0

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

**Create a sequence of dates and join this the count of transactions by date**

**transaction\_by\_day <- transactionData[order(DATE),]**

**Join to the main Table**

**Jointbydate <- cbind(countByDate,sedate )**

**Setting plot themes to format graphs**

**theme\_set(theme\_bw())**

**theme\_update(plot.title = element\_text(hjust = 0.5))**

**Plot transactions over time**

mytransOverTime <- ggplot(countByDate , aes(x = countByDate$`transactionData$DATE`, y = countByDate$n)) +

geom\_line(color = "#00AFBB")

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))

fig.align = "center"

mytransOverTime

Chart

Description automatically generated

**Filter to December and look at individual days**

filterplot <- countByDate[countByDate$`transactionData$DATE` >= "2018-12-01" & countByDate$`transactionData$DATE` <= "2018-12-31"]

ggplot( filterplot, aes(x = filterplot$`transactionData$DATE`, y = filterplot$n)) +

geom\_line(color = "#00AFBB") +

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))

Chart, line chart

Description automatically generated

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.

**Pack size**

We can work this out by taking the digits that are in PROD\_NAME

**transactionData[, PACK\_SIZE := parse\_number(PROD\_NAME)]**

Let's check if the pack sizes look sensible

**Histo<- transactionData[, .N, PACK\_SIZE][order(PACK\_SIZE)]**

**Histo**

**PACK\_SIZE N**

1: 70 1507

2: 90 3008

3: 110 22387

4: 125 1454

5: 134 25102

6: 135 3257

7: 150 40203

8: 160 2970

9: 165 15297

10: 170 19983

11: 175 66390

12: 180 1468

13: 190 2995

14: 200 4473

15: 210 6272

16: 220 1564

17: 250 3169

18: 270 6285

19: 330 12540

20: 380 6416

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

Let's plot a histogram of PACK\_SIZE since we know that it is a categorical variable and not a continuous variable even though it is numeric.

**hist(transactionData[, PACK\_SIZE])**

or we can say

**ggplot(transactionData, aes(x=PACK\_SIZE)) + geom\_histogram(color="darkblue", fill="lightblue" ,bins=20)**

Chart, histogram

Description automatically generated

**Brands**

Now to create brands, we can use the first word in PROD\_NAME to work out the brand name .Create a column which contains the brand of the product, by extracting it from the product name.

**transactionData$BRAND <- gsub("([A-Za-z]+).\*", "\\1", transactionData$PROD\_NAME)**

**transactionData[, .N, by = BRAND][order(‐N)]**

BRAND N

1: French 1418

2: NCC 1419

3: Sunbites 1432

4: GrnWves 1468

5: Woolworths 1516

6: Burger 1564

7: Snbts 1576

8: Cheetos 2927

9: Smith 2963

10: Infzns 3144

11: Dorito 3183

12: Red 4427

13: CCs 4551

14: Cheezels 4603

15: Natural 6050

16: Grain 6272

17: Tyrrells 6442

18: Twisties 9454

19: Tostitos 9471

20: Cobs 9693

21: WW 10320

22: Infuzions 11057

23: RRD 11894

24: Thins 14075

25: Doritos 22041

26: Pringles 25102

27: Smiths 27390

28: Kettle 41288

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. Let's combine these together.

**library(stringr)**

**transactionData$PROD\_Brand <- word(transactionData$PROD\_NAME, 1)**

***Checking brands and******clean brand names***

**transactionData[BRAND == "RED", BRAND := "RRD"]**

**transactionData[BRAND == "SNBTS", BRAND := "SUNBITES"]**

**transactionData[BRAND == "INFZNS", BRAND := "INFUZIONS"]**

**transactionData[BRAND == "WW", BRAND := "WOOLWORTHS"]**

**transactionData[BRAND == "Woolworths", BRAND := "WOOLWORTHS"]**

**transactionData[BRAND == "SMITH", BRAND := "SMITHS"]**

**transactionData[BRAND == "NCC", BRAND := "NATURAL"]**

**transactionData[BRAND == "DORITO", BRAND := "DORITOS"]**

**transactionData[BRAND == "GRAIN", BRAND := "GRNWVES"]**

***Check again***

**transactionData[, .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 22041

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: Pringles 25102

17: RRD 11894

18: Red 4427

19: Smith 2963

20: Smiths 27390

21: Snbts 1576

22: Sunbites 1432

23: Thins 14075

24: Tostitos 9471

25: Twisties 9454

26: Tyrrells 6442

27: WOOLWORTHS 11836

BRAND N

**Examining 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

***Examining the values of lifestage and premium\_customer***

**customerData[, .N, by = LIFESTAGE][order(N)]**

**customerData[, .N, by = PREMIUM\_CUSTOMER][order(N)]**

LIFESTAGE N

1: NEW FAMILIES 2549

2: MIDAGE SINGLES/COUPLES 7275

3: YOUNG FAMILIES 9178

4: OLDER FAMILIES 9780

5: YOUNG SINGLES/COUPLES 14441

6: OLDER SINGLES/COUPLES 14609

7: RETIREES 14805

> customerData[, .N, by = PREMIUM\_CUSTOMER][order(N)]

PREMIUM\_CUSTOMER N

1: Premium 18922

2: Budget 24470

3: Mainstream 29245

***Merge transaction data to customer data***

As the number of rows in `data` is the same as that of `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 `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.

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

***Let's also check if some customers were not matched on by checking for nulls.***

**apply(data, 2, function(x) any(is.na(x)))**

LYLTY\_CARD\_NBR DATE STORE\_NBR

FALSE FALSE FALSE

TXN\_ID PROD\_NBR PROD\_NAME

FALSE FALSE FALSE

PROD\_QTY TOT\_SALES PACK\_SIZE

FALSE FALSE FALSE

BRAND PROD\_Brand LIFESTAGE

FALSE FALSE FALSE

PREMIUM\_CUSTOMER

FALSE

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

**write this dataset into a csv file**

**write.csv(data,paste0("D:/Parisa/Internship/Quantium/","QVI\_data.csv"))**

**Data analysis on customer segments**

Now that 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 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 Let's start with calculating total sales by LIFESTAGE and PREMIUM\_CUSTOMER and plotting the split by these segments to describe which customer segment contribute most to chip sales.

***Total sales by LIFESTAGE and PREMIUM\_CUSTOMER***

**total\_sales <- data %>% group\_by(LIFESTAGE,PREMIUM\_CUSTOMER)**

**pf.total\_sales <- summarise(total\_sales,sales\_count=sum(TOT\_SALES))**

**summary(pf.total\_sales)**

***Create plot***

**library (ggmosaic)**

**install.packages("devtools")**

**install.packages("cli")**

**devtools::install\_github("haleyjeppson/ggmosaic")**

**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)**

Chart

Description automatically generated

**Total sales by LIFESTAGE and PREMIUM\_CUSTOMER**

Sales are coming mainly from Budget - older families, Mainstream - young singles/couples, and Mainstream – retirees Let's see if the higher sales are due to there being more customers who buy chips.

***Number of customers by LIFESTAGE and PREMIUM\_CUSTOMER and create a plot***

total\_sales <- data %>% group\_by(LIFESTAGE,PREMIUM\_CUSTOMER)

no\_of\_customers <- summarise(total\_sales,customer\_count = length(unique(LYLTY\_CARD\_NBR)))

summary(no\_of\_customers)

**p <- 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, '%'))))**

**p**

Chart

Description automatically generated

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 families segment. Higher sales may also be driven by more units of chips being bought per customer. Let's have a look at this next.

***Average number of units per customer by LIFESTAGE and PREMIUM\_CUSTOMER***

**units <- summarise(total\_sales, units\_count = (sum(PROD\_QTY)/uniqueN(LYLTY\_CARD\_NBR)))**

**summary(units)**

**###create plot**

**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))**

Chart, bar chart

Description automatically generated

**check <- units[order(units$units\_count, decreasing = T),]**

As shows in the plot Older families and young families in general buy more chips per customer

**Average price per unit by LIFESTAGE and PREMIUM\_CUSTOMER**

**pricePerUnit <- summarise(total\_sales,price\_per\_unit = (sum(TOT\_SALES)/sum(PROD\_QTY)))**

###plot

**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))**

Chart, bar chart

Description automatically generated

Mainstream midage 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 midage 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 Perform an independent t-test between mainstream vs premium and budget midage and young singles and couples

**pricePerUnit <- data[, price := TOT\_SALES/PROD\_QTY]**

**t.test(data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER == "Mainstream", price],data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER != "Mainstream", price], alternative = "greater")**

Welch Two Sample t-test

data: data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER == "Mainstream", price] and data[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER != "Mainstream", price]

t = 37.624, df = 54791, p-value < 2.2e-16

alternative hypothesis: true difference in means is greater than 0

95 percent confidence interval:

0.3187234 Inf

sample estimates:

mean of x mean of y

4.039786 3.706491

The t-test results in a p-value o < 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 midage 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. Let's look at Mainstream - young singles/couples. For instance, let's find out if they tend to buy a particular brand of chips.

segment1 <- data[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM\_CUSTOMER == "Mainstream",]

other <- data[!(LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM\_CUSTOMER =="Mainstream"),]

***Brand affinity compared to the rest of the population***

**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)]**

**ggplot(brand\_proportions, aes(brand\_proportions$BRAND,brand\_proportions$affinityToBrand)) + geom\_bar(stat = "identity",fill = "yellow") + labs(x = "Brand", y = "Customers Affinity to Brand", title = "Favorite brands of Customers") + theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))**

**Deep dive into Mainstream, young singles/couples**

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. |

Let's also find 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]**

**brand\_proportions <- merge(quantity\_segment1\_by\_pack, quantity\_other\_by\_pack)[, affinityTopack := targetSegment/other]**

**brand\_proportions[order(affinityTopack)]**

**ggplot(brand\_proportions, aes(brand\_proportions$PACK,brand\_proportions$affinityTopack)) + geom\_bar(stat = "identity",fill = "yellow") + labs(x = "Brand", y = "Customers Affinity to Brand", title = "Favorite brands of Customers") + theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))**

PACK\_SIZE targetSegment other affinityTopack

1: 220 0.002926156 0.006596434 0.4435967

2: 70 0.003036577 0.006322350 0.4802924

3: 200 0.008971705 0.018656115 0.4808989

4: 125 0.003008972 0.006036750 0.4984423

5: 90 0.006349206 0.012580210 0.5046980

6: 160 0.006404417 0.012372920 0.5176157

7: 180 0.003588682 0.006066692 0.5915385

8: 190 0.007481021 0.012442016 0.6012708

9: 165 0.055652174 0.062267662 0.8937572

10: 175 0.254989648 0.270006956 0.9443818

11: 150 0.157598344 0.163420656 0.9643722

12: 170 0.080772947 0.080985964 0.9973697

13: 250 0.014354727 0.012780590 1.1231662

14: 135 0.014768806 0.013075403 1.1295106

15: 210 0.029123533 0.025121265 1.1593180

16: 110 0.106280193 0.089791190 1.1836372

17: 134 0.119420290 0.100634769 1.1866703

18: 330 0.061283644 0.050161917 1.2217166

19: 380 0.032160110 0.025584213 1.2570295

20: 270 0.031828847 0.025095929 1.2682873

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

**data[PACK\_SIZE == 270, unique(PROD\_NAME)]**

"Twisties Cheese 270g" "Twisties Chicken270g"

**Conclusion/ Recommendation**

Sales have mainly been due to Budget - older families, Mainstream young singles/couples, and Mainstream - retirees shoppers. We 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, midage 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

Finally, Quantium can help considering the impact of segment and give recommendation to the category manager regarding the change of placement.

|  |  |
| --- | --- |
|  |  |
|  |  |