Data Preperation and customer analytics: Task 1

Jose A. Maldonado

March 6, 2021

### Transferring data to R

filepath = "C:/Users/Running Turtle/Desktop/Virtual Internships/Quantium/"  
transdata = fread(paste0(filepath, "QVI\_transaction\_data.csv"))  
customdata = fread(paste0(filepath, "QVI\_purchase\_behaviour.csv"))  
customdata #Checking to see the data matches the excel file.

## 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  
## ---   
## 72633: 2370651 MIDAGE SINGLES/COUPLES Mainstream  
## 72634: 2370701 YOUNG FAMILIES Mainstream  
## 72635: 2370751 YOUNG FAMILIES Premium  
## 72636: 2370961 OLDER FAMILIES Budget  
## 72637: 2373711 YOUNG SINGLES/COUPLES Mainstream

transdata #Checking to see the data matches the excel file.

## 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  
## ---   
## 264832: 43533 272 272319 270088 89  
## 264833: 43325 272 272358 270154 74  
## 264834: 43410 272 272379 270187 51  
## 264835: 43461 272 272379 270188 42  
## 264836: 43365 272 272380 270189 74  
## 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  
## ---   
## 264832: Kettle Sweet Chilli And Sour Cream 175g 2 10.8  
## 264833: Tostitos Splash Of Lime 175g 1 4.4  
## 264834: Doritos Mexicana 170g 2 8.8  
## 264835: Doritos Corn Chip Mexican Jalapeno 150g 2 7.8  
## 264836: Tostitos Splash Of Lime 175g 2 8.8

transdata$DATE = as.Date(transdata$DATE, origin = "1899-12-30") #Changing the format of the dates, excel dates start on 1899

### Exploring and validating the transactional data

str(transdata) #Seeing the structure of data set

## Classes 'data.table' and 'data.frame': 264836 obs. of 8 variables:  
## $ DATE : Date, format: "2018-10-17" "2019-05-14" ...  
## $ 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 Crinkle Cut Chips Chicken 170g" "Smiths Chip Thinly S/Cream&Onion 175g" ...  
## $ PROD\_QTY : int 2 3 2 5 3 1 1 1 1 2 ...  
## $ TOT\_SALES : num 6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...  
## - attr(\*, ".internal.selfref")=<externalptr>

head(transdata)

## 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

summary(transdata) #Summary of trans data gives us a better understanding of how the product quantity and the number of sales are correlated.

## 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.: 70021 1st Qu.: 67602   
## Median :2018-12-30 Median :130.0 Median : 130358 Median : 135138   
## Mean :2018-12-30 Mean :135.1 Mean : 135550 Mean : 135158   
## 3rd Qu.:2019-03-31 3rd Qu.:203.0 3rd Qu.: 203094 3rd Qu.: 202701   
## Max. :2019-06-30 Max. :272.0 Max. :2373711 Max. :2415841   
## PROD\_NBR PROD\_NAME PROD\_QTY TOT\_SALES   
## Min. : 1.00 Length:264836 Min. : 1.000 Min. : 1.500   
## 1st Qu.: 28.00 Class :character 1st Qu.: 2.000 1st Qu.: 5.400   
## Median : 56.00 Mode :character Median : 2.000 Median : 7.400   
## Mean : 56.58 Mean : 1.907 Mean : 7.304   
## 3rd Qu.: 85.00 3rd Qu.: 2.000 3rd Qu.: 9.200   
## Max. :114.00 Max. :200.000 Max. :650.000

transdata[, .N, PROD\_NAME] #Examining product name

## PROD\_NAME N  
## 1: Natural Chip Compny SeaSalt175g 1468  
## 2: CCs Nacho Cheese 175g 1498  
## 3: Smiths Crinkle Cut Chips Chicken 170g 1484  
## 4: Smiths Chip Thinly S/Cream&Onion 175g 1473  
## 5: Kettle Tortilla ChpsHny&Jlpno Chili 150g 3296  
## ---   
## 110: Red Rock Deli Chikn&Garlic Aioli 150g 1434  
## 111: RRD SR Slow Rst Pork Belly 150g 1526  
## 112: RRD Pc Sea Salt 165g 1431  
## 113: Smith Crinkle Cut Bolognese 150g 1451  
## 114: Doritos Salsa Mild 300g 1472

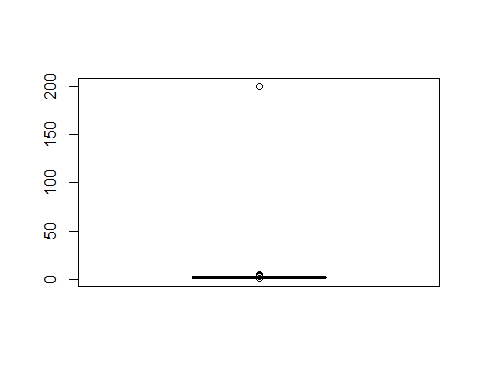
prodnameverify = data.table(unlist(strsplit(unique(transdata[,  
 PROD\_NAME]), " "))) #removed incorrect entries  
setnames(prodnameverify, 'words') #changed column name   
prodnameverify #Verify that incorrect entries were removed

## words  
## 1: Natural  
## 2: Chip  
## 3:   
## 4:   
## 5:   
## ---   
## 819: Doritos  
## 820: Salsa  
## 821: Mild  
## 822:   
## 823: 300g

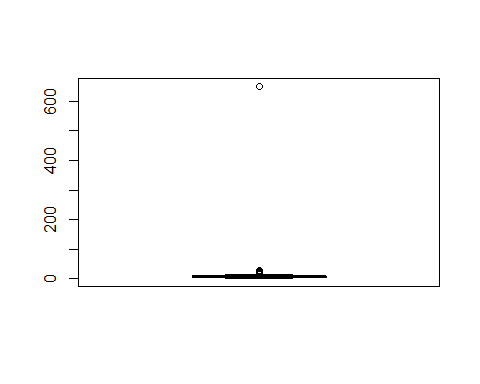
prodnameverify = prodnameverify[grepl("\\d", words) == FALSE, ] #Removing digits  
prodnameverify = prodnameverify[grepl("[:alpha:]", words), ] #Removing special characters  
verify1 = prodnameverify[, .N, words][order(N, decreasing = TRUE)] #Sorting based on frequency from highest to lowest  
transdata[, SALSA := grepl("salsa", tolower(PROD\_NAME))]  
transdata = transdata[SALSA == FALSE, ][, SALSA := NULL] #Removing salsa as we only care about chips  
  
summary(transdata) #Mainly using this to compare the quantity vs the total sales, we see that there might be potential outliers

## 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

boxplot(transdata$PROD\_QTY)



boxplot(transdata$TOT\_SALES) #Box plot lets us see potential outliers, we instantly see there are some in both the product quantity and total sales



sum(is.na(transdata))#There are no nulls in the columns.

## [1] 0

transdata[PROD\_QTY == 200, ] #Inspecting the 200 outlier, we see that happened in the same store almost a year apart.

## DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR  
## 1: 2018-08-19 226 226000 226201 4  
## 2: 2019-05-20 226 226000 226210 4  
## PROD\_NAME PROD\_QTY TOT\_SALES  
## 1: Dorito Corn Chp Supreme 380g 200 650  
## 2: Dorito Corn Chp Supreme 380g 200 650

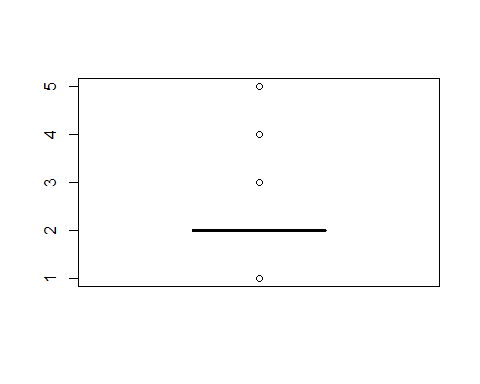
transdata[LYLTY\_CARD\_NBR == 226000, ] #Inspecting the card member to see his transactions, looks like his purchase might be more commercial then personal so removing it is best.

## DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR  
## 1: 2018-08-19 226 226000 226201 4  
## 2: 2019-05-20 226 226000 226210 4  
## PROD\_NAME PROD\_QTY TOT\_SALES  
## 1: Dorito Corn Chp Supreme 380g 200 650  
## 2: Dorito Corn Chp Supreme 380g 200 650

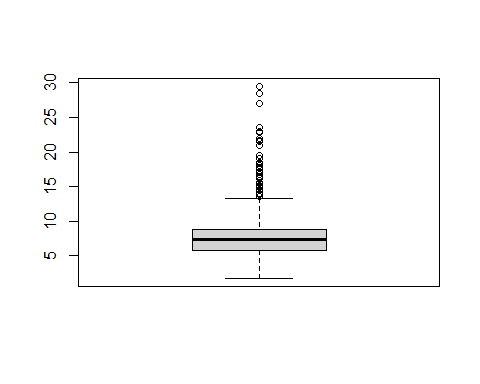
transdata = transdata[LYLTY\_CARD\_NBR != 226000,] #Removed customer  
summary(transdata) #Checking the updated summary

## DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID   
## Min. :2018-07-01 Min. : 1.0 Min. : 1000 Min. : 1   
## 1st Qu.:2018-09-30 1st Qu.: 70.0 1st Qu.: 70015 1st Qu.: 67569   
## Median :2018-12-30 Median :130.0 Median : 130367 Median : 135182   
## Mean :2018-12-30 Mean :135.1 Mean : 135530 Mean : 135130   
## 3rd Qu.:2019-03-31 3rd Qu.:203.0 3rd Qu.: 203083 3rd Qu.: 202652   
## Max. :2019-06-30 Max. :272.0 Max. :2373711 Max. :2415841   
## PROD\_NBR PROD\_NAME PROD\_QTY TOT\_SALES   
## Min. : 1.00 Length:246740 Min. :1.000 Min. : 1.700   
## 1st Qu.: 26.00 Class :character 1st Qu.:2.000 1st Qu.: 5.800   
## Median : 53.00 Mode :character Median :2.000 Median : 7.400   
## Mean : 56.35 Mean :1.906 Mean : 7.316   
## 3rd Qu.: 87.00 3rd Qu.:2.000 3rd Qu.: 8.800   
## Max. :114.00 Max. :5.000 Max. :29.500

boxplot(transdata$PROD\_QTY)



boxplot(transdata$TOT\_SALES) #Checking the updated boxplots, even though there seems to be a few outliers left this is due to the graph being zoomed in when compare to before we wouldn't consider them outliers.



transdata[, .N, by = DATE] #364 rows means there are only 364 days worth of data, we are missing one day

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

summary(transdata[, .N, by = DATE]) #summary of the number of transactions per day shows us there might be no outliers, but it is still best if we check to see which day is missing.

## 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

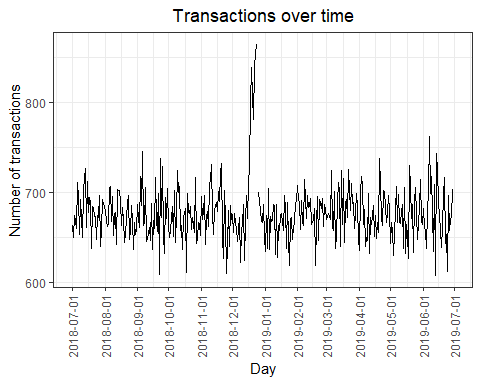
dates = data.table(seq(as.Date("2018/07/01"), as.Date("2019/06/30"), by = "day")) #Creating a sequence of dates to see which day is missing from transaction data.  
setnames(dates, "DATE") #Changed column name to match the trans data.  
dates #Checking to make sure it looks correct

## DATE  
## 1: 2018-07-01  
## 2: 2018-07-02  
## 3: 2018-07-03  
## 4: 2018-07-04  
## 5: 2018-07-05  
## ---   
## 361: 2019-06-26  
## 362: 2019-06-27  
## 363: 2019-06-28  
## 364: 2019-06-29  
## 365: 2019-06-30

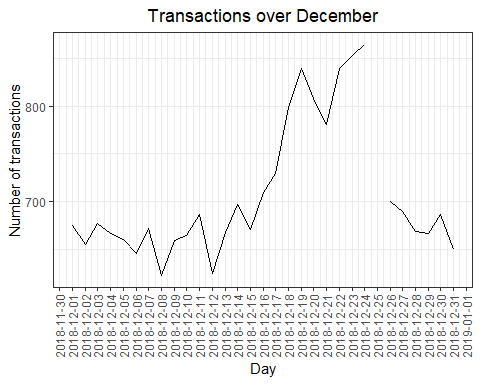
datestrans = merge(dates, transdata[, .N, by = DATE], all.x = TRUE) #Merged the original dates with the created dates  
datestrans #We see that 12/25/2018 is the missing day which no purchases were made. Makes sense since it was Christmas.

## 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  
## ---   
## 361: 2019-06-26 657  
## 362: 2019-06-27 669  
## 363: 2019-06-28 673  
## 364: 2019-06-29 703  
## 365: 2019-06-30 704

theme\_set(theme\_bw())  
theme\_update(plot.title = element\_text(hjust = 0.5)) #Setting plot themes to format graphs  
ggplot(datestrans, aes(x = DATE, y = N)) +  
 geom\_line() +  
 labs(x = "Day", y = "Number of transactions", title = "Transactions over time") + scale\_x\_date(breaks = "1 month") +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5)) #Visual representation of purchases with an increase in December and a break in there.



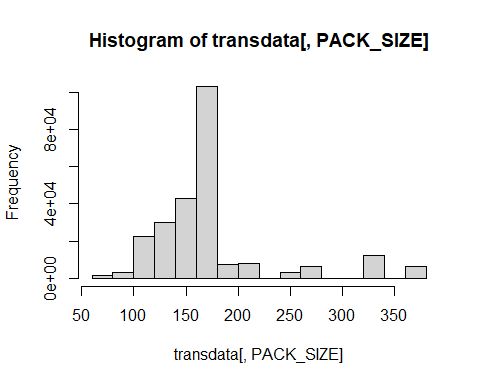
december = datestrans[month(DATE) == 12] #Isolated December  
ggplot(december, aes(x = DATE, y = N)) +  
 geom\_line() +  
 labs(x = "Day", y = "Number of transactions", title = "Transactions over December") + scale\_x\_date(breaks = "1 day") +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5)) #Zoomed in version of the graph above



transdata[, PACK\_SIZE := parse\_number(PROD\_NAME)] #Creating Pack Size  
transdata[, .N, PACK\_SIZE][order(PACK\_SIZE)] #Checking output

## 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

hist(transdata[, PACK\_SIZE]) #Histogram of the pack size



summary(transdata[, PACK\_SIZE]) #Summary of pack size gives us a better overview of the histogram as we can see min and max based on pack size

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 70.0 150.0 170.0 175.6 175.0 380.0

transdata[, BRAND := toupper(substr(PROD\_NAME, 1, regexpr(pattern = ' ', PROD\_NAME) - 1))] #Creating a column with the brands  
transdata[, .N, by = BRAND][order(-N)] #Checking the column to see if it looks reasonable while ordering it from highest to lowest based on "N"

## BRAND N  
## 1: KETTLE 41288  
## 2: SMITHS 27390  
## 3: PRINGLES 25102  
## 4: DORITOS 22041  
## 5: THINS 14075  
## 6: RRD 11894  
## 7: INFUZIONS 11057  
## 8: WW 10320  
## 9: COBS 9693  
## 10: TOSTITOS 9471  
## 11: TWISTIES 9454  
## 12: TYRRELLS 6442  
## 13: GRAIN 6272  
## 14: NATURAL 6050  
## 15: CHEEZELS 4603  
## 16: CCS 4551  
## 17: RED 4427  
## 18: DORITO 3183  
## 19: INFZNS 3144  
## 20: SMITH 2963  
## 21: CHEETOS 2927  
## 22: SNBTS 1576  
## 23: BURGER 1564  
## 24: WOOLWORTHS 1516  
## 25: GRNWVES 1468  
## 26: SUNBITES 1432  
## 27: NCC 1419  
## 28: FRENCH 1418  
## BRAND N

transdata[BRAND == "RED", BRAND := "RRD"]  
transdata[BRAND == "DORITO", BRAND := "DORITOS"]  
transdata[BRAND == "GRAIN", BRAND := "GRNWVES"]  
transdata[BRAND == "SNBTS", BRAND := "SUNBITES"]  
transdata[BRAND == "INFZNS", BRAND := "INFUZIONS"]  
transdata[BRAND == "SMITH", BRAND := "SMITHS"]  
transdata[BRAND == "WW", BRAND := "WOOLWORTHS"]  
transdata[BRAND == "NCC", BRAND := "NATURAL"] #Cleaning brand names  
transdata[, .N, by = BRAND][order(BRAND)] #Checking to see if it looks reasonable

## BRAND N  
## 1: BURGER 1564  
## 2: CCS 4551  
## 3: CHEETOS 2927  
## 4: CHEEZELS 4603  
## 5: COBS 9693  
## 6: DORITOS 25224  
## 7: FRENCH 1418  
## 8: GRNWVES 7740  
## 9: INFUZIONS 14201  
## 10: KETTLE 41288  
## 11: NATURAL 7469  
## 12: PRINGLES 25102  
## 13: RRD 16321  
## 14: SMITHS 30353  
## 15: SUNBITES 3008  
## 16: THINS 14075  
## 17: TOSTITOS 9471  
## 18: TWISTIES 9454  
## 19: TYRRELLS 6442  
## 20: WOOLWORTHS 11836

### Exploring and validating the customer data

str(customdata) #Seeing the structure of the data

## 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>

summary(customdata) #Summary of the data doesn't tell us much since lifestage, premium customer are characters and loyalty card number corresponds to a certain customer.

## 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

sum(is.na(customdata)) #There are no nulls

## [1] 0

customdata[, .N, by = LIFESTAGE][order(-N)] #Checking to see if it looks reasonable

## LIFESTAGE N  
## 1: RETIREES 14805  
## 2: OLDER SINGLES/COUPLES 14609  
## 3: YOUNG SINGLES/COUPLES 14441  
## 4: OLDER FAMILIES 9780  
## 5: YOUNG FAMILIES 9178  
## 6: MIDAGE SINGLES/COUPLES 7275  
## 7: NEW FAMILIES 2549

summary(customdata[, .N, by = LIFESTAGE])#Summary gives us a better understanding to see if there are any potential outliers

## LIFESTAGE N   
## Length:7 Min. : 2549   
## Class :character 1st Qu.: 8226   
## Mode :character Median : 9780   
## Mean :10377   
## 3rd Qu.:14525   
## Max. :14805

customdata[, .N, by = PREMIUM\_CUSTOMER][order(-N)] #Checking to see if it looks reasonable

## PREMIUM\_CUSTOMER N  
## 1: Mainstream 29245  
## 2: Budget 24470  
## 3: Premium 18922

summary(customdata[, .N, by = PREMIUM\_CUSTOMER]) #Checking for potential outliers

## PREMIUM\_CUSTOMER N   
## Length:3 Min. :18922   
## Class :character 1st Qu.:21696   
## Mode :character Median :24470   
## Mean :24212   
## 3rd Qu.:26858   
## Max. :29245

mdata = merge(transdata, customdata, all.x = TRUE) #Merging data  
sum(is.na(mdata)) #Checking to see if there are any dulls in the merged data set

## [1] 0

sum(is.na(mdata$LIFESTAGE)) #Checking to see if there are any dulls in Lifestage

## [1] 0

sum(is.na(mdata$PREMIUM\_CUSTOMER)) #Checking to see if there are any dulls in Premium Customers

## [1] 0

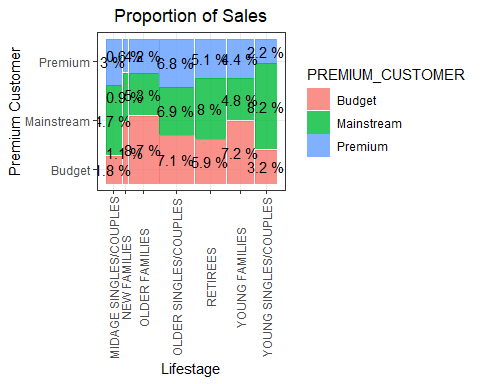
fwrite(mdata, paste0(filepath, "QVI\_mData.csv"))

### Data analysis and customer segments

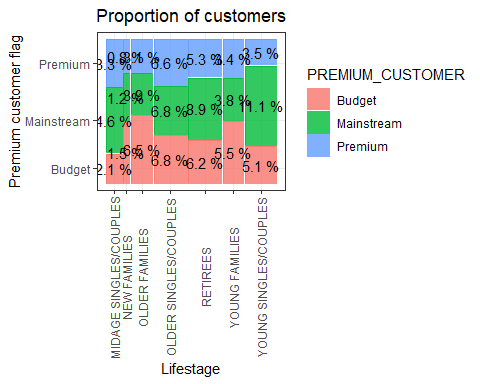
tsales = mdata[, .(SALES = sum(TOT\_SALES)), . (LIFESTAGE, PREMIUM\_CUSTOMER)] #Total sales by lifestage and premium customer  
summary(tsales) #Summary of total sales, from the looks of it we can say there aren't any outliers which makes since since we validated the data earlier

## LIFESTAGE PREMIUM\_CUSTOMER SALES   
## Length:21 Length:21 Min. : 10761   
## Class :character Class :character 1st Qu.: 54444   
## Mode :character Mode :character Median : 86338   
## Mean : 85961   
## 3rd Qu.:124649   
## Max. :156864

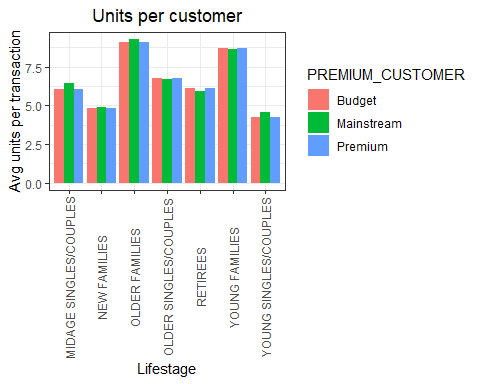
salesplot = ggplot(data = tsales) + geom\_mosaic(aes(weight = SALES, x = product(PREMIUM\_CUSTOMER, LIFESTAGE), fill = PREMIUM\_CUSTOMER)) + labs(x = "Lifestage", y = "Premium Customer", title = "Proportion of Sales") + theme(axis.text.x = element\_text(angle = 90, vjust = 0.5))  
salesplot + geom\_text(data = ggplot\_build(salesplot)$data[[1]], aes(x = (xmin + xmax)/2, y = (ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)\*100, '%'))))



numcust = mdata[, .(CUSTOMERS = uniqueN(LYLTY\_CARD\_NBR)), .(LIFESTAGE, PREMIUM\_CUSTOMER)][order(-CUSTOMERS)] #Number of customers based on Lifestage and Premium Customer  
customerplot = ggplot(data = numcust) + geom\_mosaic(aes(weight = CUSTOMERS, 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))  
customerplot + geom\_text(data = ggplot\_build(customerplot)$data[[1]], aes(x = (xmin + xmax)/2, y = (ymin + ymax)/2, label = as.character(paste(round(.wt/sum(.wt),3)\*100, '%'))))



avgunites = mdata[, .(AVG = sum(PROD\_QTY)/uniqueN(LYLTY\_CARD\_NBR)), .(LIFESTAGE, PREMIUM\_CUSTOMER)][order(-AVG)] #Average unites based on customer  
ggplot(avgunites, aes(weight = AVG, 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))



priceavg = mdata[, .(AVG = sum(TOT\_SALES)/sum(PROD\_QTY)), .(LIFESTAGE, PREMIUM\_CUSTOMER)][order(-AVG)] #Average price based on units  
ggplot(priceavg, 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") + theme(axis.title.x = element\_text(angle = 90, vjust = 0.5))



unitprice = mdata[, price := TOT\_SALES/PROD\_QTY] #  
t.test(mdata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES ") & PREMIUM\_CUSTOMER == "Mainstream", price], mdata[LIFESTAGE %in% c("YOUNG SINGLE/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER != "Mainstream", price], alternative = "greater") #The t-test results in a p-value of 2.2e-16 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.

##   
## Welch Two Sample t-test  
##   
## data: mdata[LIFESTAGE %in% c("YOUNG SINGLES/COUPLES", "MIDAGE SINGLES/COUPLES ") & PREMIUM\_CUSTOMER == "Mainstream", price] and mdata[LIFESTAGE %in% c("YOUNG SINGLE/COUPLES", "MIDAGE SINGLES/COUPLES") & PREMIUM\_CUSTOMER != "Mainstream", price]  
## t = 25.109, df = 24458, p-value < 2.2e-16  
## alternative hypothesis: true difference in means is greater than 0  
## 95 percent confidence interval:  
## 0.2853743 Inf  
## sample estimates:  
## mean of x mean of y   
## 4.065642 3.760262

c.segment1 = mdata[LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM\_CUSTOMER == "Mainstream",]  
c.segment2 = mdata[! (LIFESTAGE == "YOUNG SINGLES/COUPLES" & PREMIUM\_CUSTOMER == "Mainstream"),]  
q.segment1 = c.segment1[, sum(PROD\_QTY)]  
q.segment2 = c.segment2[, sum(PROD\_QTY)]  
q.s.brand1 = c.segment1[, .(targetSegment = sum(PROD\_QTY)/q.segment1), by = BRAND]  
q.s.brand2 = c.segment2[, .(other = sum(PROD\_QTY)/q.segment2), by = BRAND]  
b.prop = merge(q.s.brand1, q.s.brand2)[, affinityToBrand := targetSegment/other]  
b.prop[order(-affinityToBrand)]

## BRAND targetSegment other affinityToBrand  
## 1: TYRRELLS 0.031552795 0.025692464 1.2280953  
## 2: TWISTIES 0.046183575 0.037876520 1.2193194  
## 3: DORITOS 0.122760524 0.101074684 1.2145526  
## 4: KETTLE 0.197984817 0.165553442 1.1958967  
## 5: TOSTITOS 0.045410628 0.037977861 1.1957131  
## 6: PRINGLES 0.119420290 0.100634769 1.1866703  
## 7: COBS 0.044637681 0.039048861 1.1431238  
## 8: INFUZIONS 0.064679089 0.057064679 1.1334347  
## 9: THINS 0.060372671 0.056986370 1.0594230  
## 10: GRNWVES 0.032712215 0.031187957 1.0488733  
## 11: CHEEZELS 0.017971014 0.018646902 0.9637534  
## 12: SMITHS 0.096369910 0.124583692 0.7735355  
## 13: FRENCH 0.003947550 0.005758060 0.6855694  
## 14: CHEETOS 0.008033126 0.012066591 0.6657329  
## 15: RRD 0.043809524 0.067493678 0.6490908  
## 16: NATURAL 0.019599724 0.030853989 0.6352412  
## 17: CCS 0.011180124 0.018895650 0.5916771  
## 18: SUNBITES 0.006349206 0.012580210 0.5046980  
## 19: WOOLWORTHS 0.024099379 0.049427188 0.4875733  
## 20: BURGER 0.002926156 0.006596434 0.4435967

q.s.pack1 = c.segment1[, .(targetSegment = sum(PROD\_QTY)/q.segment1), by = PACK\_SIZE]  
q.s.pack2 = c.segment2[, .(other = sum(PROD\_QTY)/q.segment2), by = PACK\_SIZE]  
p.proportions = merge(q.s.pack1, q.s.pack2)[, affinityToPack := targetSegment/other]  
p.proportions[order(-affinityToPack)]

## PACK\_SIZE targetSegment other affinityToPack  
## 1: 270 0.031828847 0.025095929 1.2682873  
## 2: 380 0.032160110 0.025584213 1.2570295  
## 3: 330 0.061283644 0.050161917 1.2217166  
## 4: 134 0.119420290 0.100634769 1.1866703  
## 5: 110 0.106280193 0.089791190 1.1836372  
## 6: 210 0.029123533 0.025121265 1.1593180  
## 7: 135 0.014768806 0.013075403 1.1295106  
## 8: 250 0.014354727 0.012780590 1.1231662  
## 9: 170 0.080772947 0.080985964 0.9973697  
## 10: 150 0.157598344 0.163420656 0.9643722  
## 11: 175 0.254989648 0.270006956 0.9443818  
## 12: 165 0.055652174 0.062267662 0.8937572  
## 13: 190 0.007481021 0.012442016 0.6012708  
## 14: 180 0.003588682 0.006066692 0.5915385  
## 15: 160 0.006404417 0.012372920 0.5176157  
## 16: 90 0.006349206 0.012580210 0.5046980  
## 17: 125 0.003008972 0.006036750 0.4984423  
## 18: 200 0.008971705 0.018656115 0.4808989  
## 19: 70 0.003036577 0.006322350 0.4802924  
## 20: 220 0.002926156 0.006596434 0.4435967

mdata[PACK\_SIZE ==270, unique(PROD\_NAME)] #Mainstream youg singles/couples are 27% more likely to purchase a 270g pack of chips when compared to the rest of the population. It also seems like Twisties is the only brand that offers the 270g packs and so this results in a higher likelyhood of mainstream young singles/couples purchasing twisties.

## [1] "Twisties Cheese 270g" "Twisties Chicken270g"

#From these results we see that:  
#Mainstream young singles/couples are 26.8% more likely to purchase Tyrrells Chips compared to the rest of the population.  
#Mainstream young singles/couples are 56% more likely to buy Burger Rings compared to the rest of the populations.  
#Sales have mainly been due to budget on older families, mainstream young/couples, and mainstream retirees shoppers.  
#The high spending on chips for mainstream young singles/couples and retirees is due to more of them than other buyers.  
#Mainstream, mid-age, and young singles/couples are also more likely to pay more per pack of chips.