“BIG DATA AND ANALYTICS IN RETAIL”



Objectives

* To carry out exploratory data analysis on big-data.
* To conduct analysis on the dataset and identify the customer purchasing behaviors.
* Clustering of Products and Store number according to contribution towards sales, visit frequency and number of products purchased.
* To find out which customer type are major buyers.
* To generate useful insights from the data and provide recommendations.

DATA PREPARATION AND DESCRIPTIVE STATISTICS

# Importing data  
library(readxl)  
transaction\_data=read\_excel("C:/Users/Dell/Desktop/pro/transaction\_data.xlsx")

#Checking Structure of data  
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

dim(transaction\_data)

## [1] 264836 8

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" "Smiths Crinkle Cut Chips Chicken 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 ...

***## Here it can be observed that the DATE column is in numeric format. Also, CSV and Excel integer dates begin on 30 Dec 1899. Therefore, converting the DATE column to right format.***

transaction\_data$DATE=as.Date(transaction\_data$DATE,origin="1899-12-30")

#Summarizing Data  
head(transaction\_data)

## # A tibble: 6 x 8  
## DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR PROD\_NAME PROD\_QTY  
## <date> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 2018-10-17 1 1000 1 5 Natural Chip ~ 2  
## 2 2019-05-14 1 1307 348 66 CCs Nacho Cheese~ 3  
## 3 2019-05-20 1 1343 383 61 Smiths Crinkle C~ 2  
## 4 2018-08-17 2 2373 974 69 Smiths Chip Thin~ 5  
## 5 2018-08-18 2 2426 1038 108 Kettle Tortilla ~ 3  
## 6 2019-05-19 4 4074 2982 57 Old El Paso Sals~ 1  
## # ... with 1 more variable: TOT\_SALES <dbl>

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.: 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

#Adding Month column  
library(dplyr)

library(lubridate)

transaction\_data=transaction\_data %>%  
 mutate(MONTH=month(DATE))

head(transaction\_data)

## # A tibble: 6 x 9  
## DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR PROD\_NAME PROD\_QTY  
## <date> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>

## 1 2018-10-17 1 1000 1 5 Natural Chip ~ 2  
## 2 2019-05-14 1 1307 348 66 CCs Nacho Cheese~ 3  
## 3 2019-05-20 1 1343 383 61 Smiths Crinkle C~ 2  
## 4 2018-08-17 2 2373 974 69 Smiths Chip Thin~ 5  
## 5 2018-08-18 2 2426 1038 108 Kettle Tortilla ~ 3  
## 6 2019-05-19 4 4074 2982 57 Old El Paso Sals~ 1  
## # ... with 2 more variables: TOT\_SALES <dbl>, MONTH <dbl>

#Adding column Pack\_size to data

library(dplyr)  
library(purrr)  
library(stringr)  
transaction\_data=transaction\_data %>%  
 mutate(PACK\_SIZE = readr::parse\_number(as.character(PROD\_NAME)))  
head(transaction\_data)

## # A tibble: 6 x 10  
## DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR PROD\_NAME PROD\_QTY  
## <date> <dbl> <dbl> <dbl> <dbl> <chr> <dbl>  
## 1 2018-10-17 1 1000 1 5 Natural Chip ~ 2  
## 2 2019-05-14 1 1307 348 66 CCs Nacho Cheese~ 3  
## 3 2019-05-20 1 1343 383 61 Smiths Crinkle C~ 2  
## 4 2018-08-17 2 2373 974 69 Smiths Chip Thin~ 5  
## 5 2018-08-18 2 2426 1038 108 Kettle Tortilla ~ 3  
## 6 2019-05-19 4 4074 2982 57 Old El Paso Sals~ 1  
## # ... with 3 more variables: TOT\_SALES <dbl>, MONTH <dbl>, PACK\_SIZE <dbl>

**##Data preparation part is done now. We have added extra columns like Month, Pack size to the data which will be useful for further analysis.**

STATISTICAL ANALYSIS

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.: 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

## MONTH PACK\_SIZE   
## Min. : 1.000 Min. : 70.0   
## 1st Qu.: 4.000 1st Qu.:150.0   
## Median : 7.000 Median :170.0   
## Mean : 6.536 Mean :182.4   
## 3rd Qu.:10.000 3rd Qu.:175.0   
## Max. :12.000 Max. :380.0

***##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 products are bought in one transaction.***

#Finding and excluding outliers  
which(transaction\_data$PROD\_QTY==200)

## [1] 69763 69764

transaction\_data[c(69763,69764),]

# A tibble: 2 x 10

DATE STORE\_NBR LYLTY\_CARD\_NBR TXN\_ID PROD\_NBR PROD\_NAME PROD\_QTY TOT\_SALES

<date> <dbl> <dbl> <dbl> <dbl> <chr> <dbl> <dbl>

1 2018-08-19 226 226000 226201 4 Dorito Corn~ 200 650

2 2019-05-20 226 226000 226210 4 Dorito Corn~ 200 650

***## There are two transactions where 200 products are bought in one transaction and both of these transactions were done by the same customer.***

transaction\_data=transaction\_data[-(69763:69764),]  
dim(transaction\_data)

## [1] 264834 10

***## 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. We’ll remove this loyalty card number from data frame.***

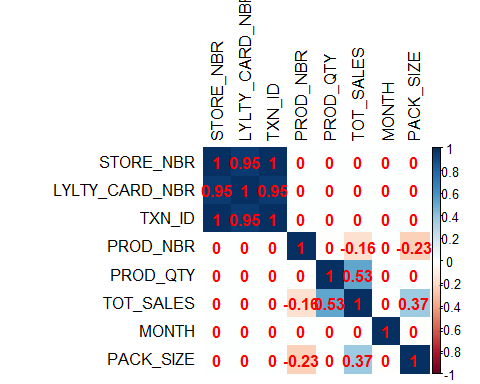
str(transaction\_data)

## tibble [264,834 x 10] (S3: tbl\_df/tbl/data.frame)  
## $ DATE : Date[1:264834], format: "2018-10-17" "2019-05-14" ...  
## $ STORE\_NBR : num [1:264834] 1 1 1 2 2 4 4 4 5 7 ...  
## $ LYLTY\_CARD\_NBR: num [1:264834] 1000 1307 1343 2373 2426 ...  
## $ TXN\_ID : num [1:264834] 1 348 383 974 1038 ...  
## $ PROD\_NBR : num [1:264834] 5 66 61 69 108 57 16 24 42 52 ...  
## $ PROD\_NAME : chr [1:264834] "Natural Chip Compny SeaSalt175g" "CCs Nacho Cheese 175g" "Smiths Crinkle Cut Chips Chicken 170g" "Smiths Chip Thinly S/Cream&Onion 175g" ...  
## $ PROD\_QTY : num [1:264834] 2 3 2 5 3 1 1 1 1 2 ...  
## $ TOT\_SALES : num [1:264834] 6 6.3 2.9 15 13.8 5.1 5.7 3.6 3.9 7.2 ...  
## $ MONTH : num [1:264834] 10 5 5 8 8 5 5 5 8 8 ...  
## $ PACK\_SIZE : num [1:264834] 175 175 170 175 150 300 330 210 150 210 ...

***## Now we find that our data is in better format and good for further analysis.***

#To see Correlation between variables  
library(corrplot)

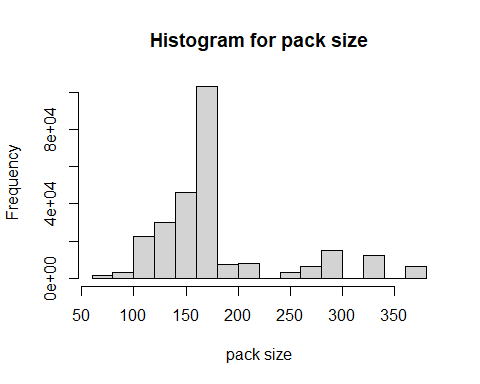
correlation=cor(transaction\_data[,-c(1,6)])  
corrplot(correlation, method="color", addCoef.col= "red",tl.col="black")



**##From above correlation plot it can be observed that store number and transaction id are completely correlated. Also, Loyalty Card number and Store number are highly correlated. While other correlations may be neglected as they are not much significant.**

#Finding which preferred pack size

hist(transaction\_data$PACK\_SIZE)



y=table(transaction\_data$PACK\_SIZE)  
names(y)[which(y==max(y))]

## [1] "175"

***## Pack sizes created look reasonable. From Histogram we can conclude the pack size of 175g is most preferred by customers.***

# Cleaning and Analyzing Product name column

name=data.frame(table((transaction\_data$PROD\_NAME)))  
library(wordcloud2)

library(RColorBrewer)  
library(tm)

getwd()

## [1] "C:/Users/Dell/Desktop/pro"

library(readr)  
write\_tsv(name, file = "name.txt")  
text=readLines(file.choose())  
docs=Corpus(VectorSource(text))  
library(dplyr)

docs=docs %>%  
 tm\_map(removeNumbers) %>%  
 tm\_map(removePunctuation) %>%

tm\_map(stripWhitespace)

docs=tm\_map(docs, content\_transformer(tolower))

docs=tm\_map(docs, removeWords, stopwords("english"))

dtm=TermDocumentMatrix(docs)   
matrix=as.matrix(dtm)   
words=sort(rowSums(matrix),decreasing=TRUE)   
df=data.frame(word = names(words),freq=words)  
set.seed(678)  
wordcloud2(data=df,size=1.2, color='random-dark')



**##From above word cloud we can say that the product “Chips” is most purchased product. The popular brands are “Kettle” brand followed by “RRD” brand.**

***##Most frequently bought products***

count(transaction\_data, 'PROD\_NAME')

## PROD\_NAME freq  
## 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  
## 11 Cobs Popd Swt/Chlli &Sr/Cream Chips 110g 3269  
## 12 Dorito Corn Chp Supreme 380g 3185  
## 13 Doritos Cheese Supreme 330g 3052  
## 14 Doritos Corn Chip Mexican Jalapeno 150g 3204  
## 15 Doritos Corn Chip Southern Chicken 150g 3172  
## 16 Doritos Corn Chips Cheese Supreme 170g 3217  
## 17 Doritos Corn Chips Nacho Cheese 170g 3160  
## 18 Doritos Corn Chips Original 170g 3121  
## 19 Doritos Mexicana 170g 3115  
## 20 Doritos Salsa Medium 300g 1449  
## 21 Doritos Salsa Mild 300g 1472  
## 22 French Fries Potato Chips 175g 1418  
## 23 Grain Waves Sweet Chilli 210g 3167  
## 24 Grain Waves Sour Cream&Chives 210G 3105  
## 25 GrnWves Plus Btroot & Chilli Jam 180g 1468  
## 26 Infuzions BBQ Rib Prawn Crackers 110g 3174  
## 27 Infuzions Mango Chutny Papadums 70g 1507  
## 28 Infuzions SourCream&Herbs Veg Strws 110g 3134  
## 29 Infuzions Thai SweetChili PotatoMix 110g 3242  
## 30 Infzns Crn Crnchers Tangy Gcamole 110g 3144  
## 31 Kettle 135g Swt Pot Sea Salt 3257  
## 32 Kettle Chilli 175g 3038  
## 33 Kettle Honey Soy Chicken 175g 3148  
## 34 Kettle Mozzarella Basil & Pesto 175g 3304  
## 35 Kettle Original 175g 3159  
## 36 Kettle Sea Salt And Vinegar 175g 3173  
## 37 Kettle Sensations BBQ&Maple 150g 3083  
## 38 Kettle Sensations Camembert & Fig 150g 3219  
## 39 Kettle Sensations Siracha Lime 150g 3127  
## 40 Kettle Sweet Chilli And Sour Cream 175g 3200  
## 41 Kettle Tortilla ChpsBtroot&Ricotta 150g 3146  
## 42 Kettle Tortilla ChpsFeta&Garlic 150g 3138  
## 43 Kettle Tortilla ChpsHny&Jlpno Chili 150g 3296  
## 44 Natural Chip Compny SeaSalt175g 1468  
## 45 Natural Chip Co Tmato Hrb&Spce 175g 1572  
## 46 Natural ChipCo Hony Soy Chckn175g 1460  
## 47 Natural ChipCo Sea Salt & Vinegr 175g 1550  
## 48 NCC Sour Cream & Garden Chives 175g 1419  
## 49 Old El Paso Salsa Dip Chnky Tom Ht300g 3125  
## 50 Old El Paso Salsa Dip Tomato Med 300g 3114  
## 51 Old El Paso Salsa Dip Tomato Mild 300g 3085  
## 52 Pringles Barbeque 134g 3210  
## 53 Pringles Chicken Salt Crips 134g 3104  
## 54 Pringles Mystery Flavour 134g 3114  
## 55 Pringles Original Crisps 134g 3157  
## 56 Pringles Slt Vingar 134g 3095  
## 57 Pringles SourCream Onion 134g 3162  
## 58 Pringles Sthrn FriedChicken 134g 3083  
## 59 Pringles Sweet&Spcy BBQ 134g 3177  
## 60 Red Rock Deli Chikn&Garlic Aioli 150g 1434  
## 61 Red Rock Deli Sp Salt & Truffle 150G 1498  
## 62 Red Rock Deli SR Salsa & Mzzrlla 150g 1458  
## 63 Red Rock Deli Thai Chilli&Lime 150g 1495  
## 64 RRD Chilli& Coconut 150g 1506  
## 65 RRD Honey Soy Chicken 165g 1513  
## 66 RRD Lime & Pepper 165g 1473  
## 67 RRD Pc Sea Salt 165g 1431  
## 68 RRD Salt & Vinegar 165g 1474  
## 69 RRD SR Slow Rst Pork Belly 150g 1526  
## 70 RRD Steak & Chimuchurri 150g 1455  
## 71 RRD Sweet Chilli & Sour Cream 165g 1516  
## 72 Smith Crinkle Cut Bolognese 150g 1451  
## 73 Smith Crinkle Cut Mac N Cheese 150g 1512  
## 74 Smiths Chip Thinly Cut Original 175g 1614  
## 75 Smiths Chip Thinly CutSalt/Vinegr175g 1440  
## 76 Smiths Chip Thinly S/Cream&Onion 175g 1473  
## 77 Smiths Crinkle Original 330g 3142  
## 78 Smiths Crinkle Chips Salt & Vinegar 330g 3197  
## 79 Smiths Crinkle Cut Chips Barbecue 170g 1489  
## 80 Smiths Crinkle Cut Chips Chicken 170g 1484  
## 81 Smiths Crinkle Cut Chips Chs&Onion170g 1481  
## 82 Smiths Crinkle Cut Chips Original 170g 1461  
## 83 Smiths Crinkle Cut French OnionDip 150g 1438  
## 84 Smiths Crinkle Cut Salt & Vinegar 170g 1455  
## 85 Smiths Crinkle Cut Snag&Sauce 150g 1503  
## 86 Smiths Crinkle Cut Tomato Salsa 150g 1470  
## 87 Smiths Crnkle Chip Orgnl Big Bag 380g 3233  
## 88 Smiths Thinly Swt Chli&S/Cream175G 1461  
## 89 Smiths Thinly Cut Roast Chicken 175g 1519  
## 90 Snbts Whlgrn Crisps Cheddr&Mstrd 90g 1576  
## 91 Sunbites Whlegrn Crisps Frch/Onin 90g 1432  
## 92 Thins Chips Originl saltd 175g 1441  
## 93 Thins Chips Light& Tangy 175g 3188  
## 94 Thins Chips Salt & Vinegar 175g 3103  
## 95 Thins Chips Seasonedchicken 175g 3114  
## 96 Thins Potato Chips Hot & Spicy 175g 3229  
## 97 Tostitos Lightly Salted 175g 3074  
## 98 Tostitos Smoked Chipotle 175g 3145  
## 99 Tostitos Splash Of Lime 175g 3252  
## 100 Twisties Cheese 270g 3115  
## 101 Twisties Cheese Burger 250g 3169  
## 102 Twisties Chicken270g 3170  
## 103 Tyrrells Crisps Ched & Chives 165g 3268  
## 104 Tyrrells Crisps Lightly Salted 165g 3174  
## 105 Woolworths Cheese Rings 190g 1516  
## 106 Woolworths Medium Salsa 300g 1430  
## 107 Woolworths Mild Salsa 300g 1491  
## 108 WW Crinkle Cut Chicken 175g 1467  
## 109 WW Crinkle Cut Original 175g 1410  
## 110 WW D/Style Chip Sea Salt 200g 1469  
## 111 WW Original Corn Chips 200g 1495  
## 112 WW Original Stacked Chips 160g 1487  
## 113 WW Sour Cream &OnionStacked Chips 160g 1483  
## 114 WW Supreme Cheese Corn Chips 200g 150

**## The most frequently bought Products were, Kettle Mozzarella Basil & Pesto *175g bought 3304 times, Kettle Tortilla ChpsHny&Jlpno Chili 150g 3296 times and Tyrrells Crisps Ched & Chives 165g, 3268 times.***

**## Products contributing total sales**

sale\_store=transaction\_data %>%  
 group\_by(PROD\_NAME) %>%  
 summarize(tot\_sales=sum(TOT\_SALES))  
  
product\_df=cbind(unique(transaction\_data$PROD\_NAME),unique(transaction\_data$PROD\_NBR),sale\_store$tot\_sales)  
colnames(product\_df)=c("Prod\_name","Prod\_num","Total\_sales") head(product\_df)

## Prod\_name Prod\_num Total\_sales  
## [1,] "Natural Chip Compny SeaSalt175g" "5" "6831"   
## [2,] "CCs Nacho Cheese 175g" "66" "5961.9"   
## [3,] "Smiths Crinkle Cut Chips Chicken 170g" "61" "6048"   
## [4,] "Smiths Chip Thinly S/Cream&Onion 175g" "69" "6069"   
## [5,] "Kettle Tortilla ChpsHny&Jlpno Chili 150g" "108" "9243.3"   
## [6,] "Old El Paso Salsa Dip Tomato Mild 300g" "57" "7641.2"

product\_grp=cbind(unique(transaction\_data$PROD\_NBR),sale\_store$tot\_sales)  
product\_grp

## [,1] [,2]  
## [1,] 5 6831.0  
## [2,] 66 5961.9  
## [3,] 61 6048.0  
## [4,] 69 6069.0  
## [5,] 108 9243.3  
## [6,] 57 7641.2  
## [7,] 16 34296.9  
## [8,] 24 5733.0  
## [9,] 42 23852.6  
## [10,] 52 22944.4  
## [11,] 114 23772.8  
## [12,] 15 40352.0  
## [13,] 92 33390.6  
## [14,] 44 23887.5  
## [15,] 54 23735.4  
## [16,] 94 27183.2  
## [17,] 98 26562.8  
## [18,] 93 26228.4  
## [19,] 56 26290.0  
## [20,] 7 7111.0  
## [21,] 31 7150.0  
## [22,] 32 7929.0  
## [23,] 111 21700.8  
## [24,] 46 21348.0  
## [25,] 13 8568.4  
## [26,] 99 23111.6  
## [27,] 26 6852.0  
## [28,] 64 22701.2  
## [29,] 22 23582.8  
## [30,] 48 22800.0  
## [31,] 37 26090.4  
## [32,] 36 31271.4  
## [33,] 51 32578.2  
## [34,] 107 34457.4  
## [35,] 106 32740.2  
## [36,] 4 32589.0  
## [37,] 113 27186.0  
## [38,] 45 28308.4  
## [39,] 39 27567.8  
## [40,] 102 33031.8  
## [41,] 104 27770.2  
## [42,] 3 27627.6  
## [43,] 82 29021.4  
## [44,] 88 8331.0  
## [45,] 40 8934.0  
## [46,] 73 8274.0  
## [47,] 87 8733.0  
## [48,] 84 8046.0  
## [49,] 70 30513.3  
## [50,] 89 30237.9  
## [51,] 101 30033.9  
## [52,] 63 22614.4  
## [53,] 25 22063.1  
## [54,] 47 21937.3  
## [55,] 71 22355.4  
## [56,] 65 21963.2  
## [57,] 33 22410.9  
## [58,] 35 21833.7  
## [59,] 12 22477.5  
## [60,] 8 7384.5  
## [61,] 75 7678.8  
## [62,] 100 7438.5  
## [63,] 29 7589.7  
## [64,] 59 7654.5  
## [65,] 30 8649.0  
## [66,] 81 8379.0  
## [67,] 67 8106.0  
## [68,] 110 8364.0  
## [69,] 28 7778.7  
## [70,] 2 7449.3  
## [71,] 14 8574.0  
## [72,] 77 7118.8  
## [73,] 17 7464.6  
## [74,] 83 9135.0  
## [75,] 68 8196.0  
## [76,] 96 8313.0  
## [77,] 79 34302.6  
## [78,] 23 34804.2  
## [79,] 50 8125.8  
## [80,] 78 8183.8  
## [81,] 1 8111.3  
## [82,] 86 8001.1  
## [83,] 53 7046.0  
## [84,] 72 7986.6  
## [85,] 74 7412.6  
## [86,] 76 7168.2  
## [87,] 9 36367.6  
## [88,] 91 8325.0  
## [89,] 105 8598.0  
## [90,] 90 5076.2  
## [91,] 109 4600.2  
## [92,] 27 8999.1  
## [93,] 62 20113.5  
## [94,] 112 19575.6  
## [95,] 55 19753.8  
## [96,] 18 20410.5  
## [97,] 34 25885.2  
## [98,] 49 26474.8  
## [99,] 60 27429.6  
## [100,] 38 27572.4  
## [101,] 103 26096.7  
## [102,] 85 27853.0  
## [103,] 95 26149.2  
## [104,] 97 25498.2  
## [105,] 20 5169.6  
## [106,] 19 4050.0  
## [107,] 21 4234.5  
## [108,] 6 4702.2  
## [109,] 80 4532.2  
## [110,] 58 5249.7  
## [111,] 10 5367.5  
## [112,] 11 5323.8  
## [113,] 43 5323.8  
## [114,] 41 5390.3

**## The products having product number 15, 9, 23 were contributing maximum toward total sales.**

***## Clustering the products in accordance to contribution towards sales, frequency and quantity of product bought.***

sale\_tab=transaction\_data %>%  
 group\_by(PROD\_NBR) %>%  
 summarize(sale=sum(TOT\_SALES),  
 qty=sum(PROD\_QTY))  
  
product\_df=cbind(sale\_tab, prod\_frq$Freq)  
colnames(product\_df)=c('pr\_nbr','sale','qty','freq'); head(scale(product\_df))

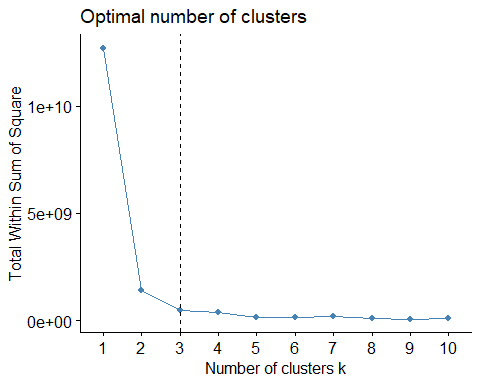
## pr\_nbr sale qty freq  
## 1 -1.709376 -0.8474119 -0.9928361 -0.9863006  
## 2 -1.679122 0.5726729 0.9795338 0.9883750  
## 3 -1.648868 1.0867116 1.0520653 1.0593215  
## 4 -1.618613 2.2408647 1.2666124 1.0191185  
## 5 -1.588359 -0.8277474 -1.0080738 -1.0111318  
## 6 -1.558104 -0.8231475 -0.9983216 -1.0052196

***## Elbow diagram to find optimal value of K***

library(cluster)

library(factoextra)

fviz\_nbclust(product\_df, kmeans, method = "wss") + geom\_vline(xintercept = 3, linetype = 2)



**## The optimum number of clusters obtained is 3.**

**# Forming Clusters**

clusters=kmeans(product\_df, 3)  
clusters

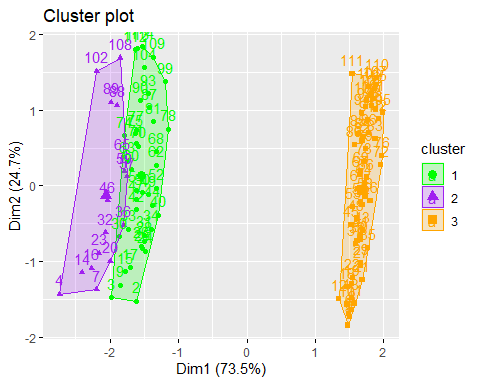
## K-means clustering with 3 clusters of sizes 41, 16, 57  
##   
## Cluster means:  
## pr\_nbr sale qty freq  
## 1 57.36585 24261.461 6054.854 3163.927  
## 2 47.87500 33124.275 6075.688 3162.500  
## 3 60.29825 7187.837 2801.123 1482.719  
##   
## Clustering vector:

## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 3 1 1 2 3 3 2 3 1 3 3 3 3 2 1 2 1 3 3 2   
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40   
## 3 3 2 1 1 1 3 1 3 1 1 2 1 1 3 2 3 3 3 1   
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60   
## 3 1 3 1 3 2 1 3 1 1 1 1 3 3 3 3 2 3 2 1   
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
## 3 1 1 3 2 3 3 1 3 1 1 3 3 1 1 3 1 1 3 3   
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100   
## 1 3 3 3 3 3 1 2 2 1 3 3 1 3 3 3 3 3 1 3   
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114   
## 3 2 3 1 3 3 3 2 1 3 3 1 1 1   
##   
## Within cluster sum of squares by cluster:  
## [1] 267793026 116549856 112427224  
## (between\_SS / total\_SS = 96.1 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

fviz\_cluster(clusters, data =as.data.frame(product\_df),  
 palette = c("green", "purple", "orange"),   
 )

**##Here cluster 2 obtained is the best cluster as it contributes maximum to the sales and has maximum frequency followed by cluster 1 and cluster 3 having comparatively minimum values.**

#**Visualizing Clusters**



#Exploring Consumer data  
library(readr)  
library(dplyr)  
Customer\_data=read\_csv("C:/Users/Dell/Desktop/pro/purchase\_data.csv")

##   
## -- Column specification --------------------------------------------------------  
## cols(  
## LYLTY\_CARD\_NBR = col\_double(),  
## LIFESTAGE = col\_character(),  
## PREMIUM\_CUSTOMER = col\_character()  
## )

head(Customer\_data)

## # A tibble: 6 x 3  
## LYLTY\_CARD\_NBR LIFESTAGE PREMIUM\_CUSTOMER  
## <dbl> <chr> <chr>   
## 1 1000 YOUNG SINGLES/COUPLES Premium   
## 2 1002 YOUNG SINGLES/COUPLES Mainstream   
## 3 1003 YOUNG FAMILIES Budget   
## 4 1004 OLDER SINGLES/COUPLES Mainstream   
## 5 1005 MIDAGE SINGLES/COUPLES Mainstream   
## 6 1007 YOUNG SINGLES/COUPLES Budget

dim(Customer\_data)

## [1] 72637 3

table(Customer\_data$LIFESTAGE)

##   
## MIDAGE SINGLES/COUPLES NEW FAMILIES OLDER FAMILIES   
## 7275 2549 9780   
## OLDER SINGLES/COUPLES RETIREES YOUNG FAMILIES   
## 14609 14805 9178   
## YOUNG SINGLES/COUPLES   
## 14441

table(Customer\_data$PREMIUM\_CUSTOMER)

##   
## Budget Mainstream Premium   
## 24470 29245 18922

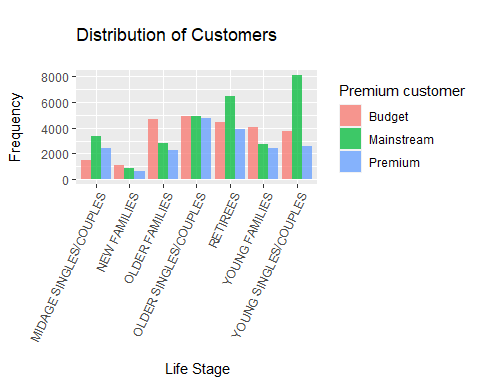
table(Customer\_data$LIFESTAGE,Customer\_data$PREMIUM\_CUSTOMER)

##   
## Budget Mainstream Premium  
## MIDAGE SINGLES/COUPLES 1504 3340 2431  
## NEW FAMILIES 1112 849 588  
## OLDER FAMILIES 4675 2831 2274  
## OLDER SINGLES/COUPLES 4929 4930 4750  
## RETIREES 4454 6479 3872  
## YOUNG FAMILIES 4017 2728 2433  
## YOUNG SINGLES/COUPLES 3779 8088 2574

***## It is clear that maximum customers are of mainstream with life stage young singles/Couples. We can visualize the above table graphically as following.***

info=data.frame(table(Customer\_data$LIFESTAGE,Customer\_data$PREMIUM\_CUSTOMER))  
colnames(info)=c("Life stage","Premium customer","Freq")

library(ggplot2)  
ggplot(data =info, aes(x=`Life stage`, y=Freq, fill=`Premium customer`)) +  
 geom\_bar(stat = "identity", position = position\_dodge(), alpha = 0.75)+  
 labs(x="\n Life Stage", y="Frequency\n", title="\nDistribution of Customers\n")+  
 theme(axis.text.x = element\_text(angle =65, hjust = 1))



***## It can be concluded that the most of the customers are of Mainstream type. From those Young singles/Couples and Retirees being maximum. Also new Families being least customers.***

#Combining transaction and customer data set.  
trans\_data=transaction\_data %>%  
 group\_by(LYLTY\_CARD\_NBR) %>%  
 summarize(Prod\_qty=sum(PROD\_QTY),  
 Tot\_sales=sum(TOT\_SALES),  
 Store\_Nbr=mean(STORE\_NBR),  
 Pack\_size=sum(PACK\_SIZE));

head(trans\_data)

## # A tibble: 6 x 5  
## LYLTY\_CARD\_NBR Prod\_qty Tot\_sales Store\_Nbr Pack\_size  
## <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 1000 2 6 1 175  
## 2 1002 1 2.7 1 150  
## 3 1003 2 6.6 1 385  
## 4 1004 1 1.9 1 160  
## 5 1005 1 2.8 1 165  
## 6 1007 2 6.5 1 260

dim(trans\_data)

## [1] 72636 5

comb\_data=data.frame(inner\_join(trans\_data,Customer\_data,by="LYLTY\_CARD\_NBR"))  
head(comb\_data)

## LYLTY\_CARD\_NBR Prod\_qty Tot\_sales Store\_Nbr Pack\_size LIFESTAGE  
## 1 1000 2 6.0 1 175 YOUNG SINGLES/COUPLES  
## 2 1002 1 2.7 1 150 YOUNG SINGLES/COUPLES  
## 3 1003 2 6.6 1 385 YOUNG FAMILIES  
## 4 1004 1 1.9 1 160 OLDER SINGLES/COUPLES  
## 5 1005 1 2.8 1 165 MIDAGE SINGLES/COUPLES  
## 6 1007 2 6.5 1 260 YOUNG SINGLES/COUPLES

## PREMIUM\_CUSTOMER  
## 1 Premium  
## 2 Mainstream  
## 3 Budget  
## 4 Mainstream  
## 5 Mainstream  
## 6 Budget

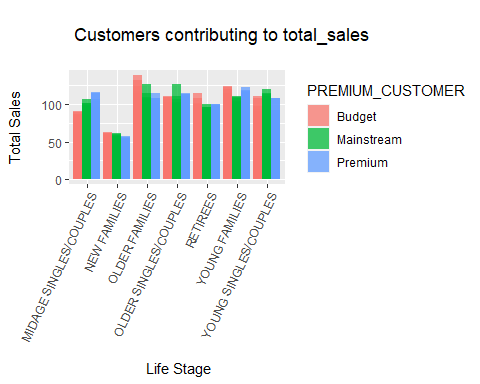
dim(comb\_data)

## [1] 72636 7

summary(comb\_data)

## LYLTY\_CARD\_NBR Prod\_qty Tot\_sales Store\_Nbr   
## Min. : 1000 Min. : 1.000 Min. : 1.50 Min. : 1.0   
## 1st Qu.: 66202 1st Qu.: 2.000 1st Qu.: 9.10 1st Qu.: 66.0   
## Median : 134040 Median : 6.000 Median : 21.70 Median :133.0   
## Mean : 136185 Mean : 6.949 Mean : 26.61 Mean :135.1   
## 3rd Qu.: 203374 3rd Qu.:10.000 3rd Qu.: 40.00 3rd Qu.:203.0   
## Max. :2373711 Max. :36.000 Max. :138.60 Max. :272.0   
## Pack\_size LIFESTAGE PREMIUM\_CUSTOMER   
## Min. : 70.0 Length:72636 Length:72636   
## 1st Qu.: 280.0 Class :character Class :character   
## Median : 520.0 Mode :character Mode :character   
## Mean : 665.1   
## 3rd Qu.: 965.0   
## Max. :3395.0

library(ggplot2)  
ggplot(data =comb\_data, aes(x=`LIFESTAGE`, y=`Tot\_sales`, fill=`PREMIUM\_CUSTOMER`)) +  
 geom\_bar(stat = "identity", position = position\_dodge(), alpha = 0.75)+  
 labs(x="\n Life Stage", y="Total Sales\n", title="\n Customers contributing to total\_sales\n")+theme(axis.text.x = element\_text(angle =65, hjust = 1))



***##It can be concluded from above graph that Budget, Mainstream and Premium customers within each life stage is almost similar.*** ***Older families contributing most towards total sales followed by young families and Retirees. Again, new families being least.***

#Checking total sales  
total\_sales=sum(transaction\_data$TOT\_SALES);total\_sales

## [1] 1933115

***##The total sales from last year is 1933115 units.***

#Checking visit frequency of customers by Store number

d\_table=data.frame(sort(table(transaction\_data$STORE\_NBR)))

f=d\_table$Freq

d\_table %>% mutate(cum\_per=(cumsum(f)/sum(f)\*100))

Var1 Freq cum\_per

1 76 1 3.775922e-04

2 92 1 7.551843e-04

3 11 2 1.510369e-03

4 31 2 2.265553e-03

5 206 2 3.020737e-03

6 211 2 3.775922e-03

7 252 2 4.531106e-03

8 85 3 5.663883e-03

9 193 3 6.796659e-03

10 117 46 2.416590e-02

11 198 47 4.191273e-02

12 258 47 5.965956e-02

13 146 56 8.080472e-02

14 177 56 1.019499e-01

15 44 58 1.238502e-01

16 244 59 1.461282e-01

17 140 61 1.691613e-01

18 263 61 1.921944e-01

19 42 64 2.163603e-01

20 99 64 2.405262e-01

21 14 66 2.654473e-01

22 267 66 2.903684e-01

23 158 67 3.156671e-01

24 161 67 3.409657e-01

25 218 67 3.662644e-01

26 224 67 3.915631e-01

27 139 68 4.172393e-01

28 159 68 4.429156e-01

29 204 68 4.685919e-01

30 127 72 4.957785e-01

31 135 73 5.233427e-01

………………

………………

………………

239 72 1628 7.864150e+01

240 270 1630 7.925697e+01

241 157 1631 7.987283e+01

242 69 1633 8.048943e+01

243 223 1636 8.110718e+01

244 26 1640 8.172643e+01

245 94 1641 8.234606e+01

246 130 1641 8.296568e+01

247 259 1642 8.358569e+01

248 81 1649 8.420834e+01

249 112 1651 8.483174e+01

250 201 1654 8.545628e+01

251 203 1661 8.608346e+01

252 71 1676 8.671631e+01

253 4 1678 8.734991e+01

254 168 1682 8.798502e+01

255 199 1687 8.862202e+01

256 179 1702 8.926468e+01

257 133 1703 8.990772e+01

258 152 1707 9.055227e+01

259 100 1711 9.119833e+01

260 128 1714 9.184552e+01

261 63 1742 9.250329e+01

262 58 1743 9.316143e+01

263 156 1747 9.382108e+01

264 213 1747 9.448074e+01

265 230 1751 9.514190e+01

266 40 1764 9.580797e+01

267 43 1771 9.647669e+01

268 237 1785 9.715069e+01

269 165 1819 9.783753e+01

270 93 1832 9.852928e+01

271 88 1873 9.923651e+01

272 226 2022 1.000000e+02

***##******The stores with store number 226, 88, 93, 165 and 237 are the most visited. It can be observed that some stores contribute very least visited. The first 30 stores are visited less than 0.5%. Therefore, some action may be taken to improve the performance of those stores.***

***## Most frequently visited stores***

str\_frq=data.frame(table(transaction\_data$STORE\_NBR))  
sr\_fr = str\_frq[order(str\_frq$Freq),]  
tail(sr\_fr)

## Var1 Freq  
## 43 43 1771  
## 237 237 1785  
## 165 165 1819  
## 93 93 1832  
## 88 88 1873  
## 226 226 2022

***## Store number 43, 237, 165, 93, 88, 226 are most frequently visited stores and also contributed the most toward sales.***

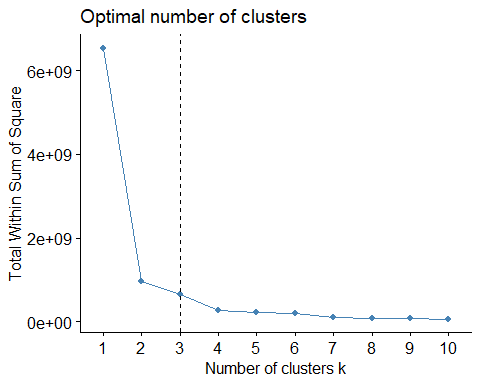
***## Data Preparation for Clustering***

library(dplyr)  
sale\_tab1=transaction\_data %>%  
 group\_by(STORE\_NBR) %>%  
 summarize(sale=sum(TOT\_SALES),  
 qty=sum(PROD\_QTY))  
  
store\_df=cbind(sale\_tab1, str\_frq$Freq)  
colnames(store\_df)=c('str\_nbr','sale','qty','freq'); head(scale(store\_df))

## str\_nbr sale qty freq  
## 1 -1.722519 -1.0002138 -0.9467339 -0.6816428  
## 2 -1.709807 -1.0824234 -1.0616813 -0.7956294  
## 3 -1.697095 1.2063547 0.8686132 0.8988578  
## 4 -1.684383 1.5975180 1.1978552 1.1982852  
## 5 -1.671670 0.5064391 0.6953711 0.6589756  
## 6 -1.658958 -0.9384612 -0.8941866 -0.7547984

***## Finding optimal number of clusters using elbow method***

library(cluster)  
library(factoextra)  
fviz\_nbclust(store\_df, kmeans, method = "wss") + geom\_vline(xintercept = 3, linetype = 2)



***## Optimal number of clusters is 3.***

***## Formation of Clusters***

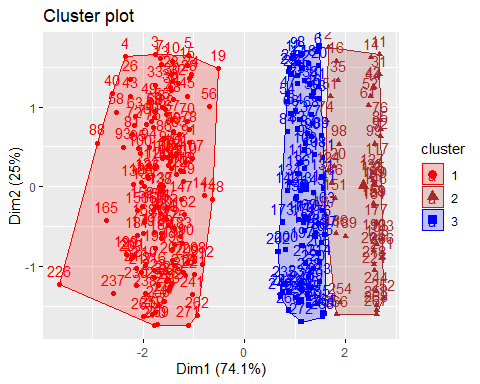
clusters=kmeans(store\_df, 3)  
clusters

## K-means clustering with 3 clusters of sizes 135, 50, 87  
##   
## Cluster means:  
## str\_nbr sale qty freq  
## 1 133.4000 11481.430 3022.1852 1533.800  
## 2 139.9400 736.210 210.9600 172.320  
## 3 139.3333 3995.533 995.1839 565.023  
##   
## Clustering vector:  
## 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20   
## 3 2 1 1 1 3 1 3 3 1 2 3 1 2 1 2 3 3 1 3   
## 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40   
## 3 3 1 1 3 1 3 1 3 1 2 1 1 3 2 1 3 3 1 1   
## 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60   
## 3 2 1 2 1 3 3 1 1 3 2 2 3 3 1 1 1 1 1 1   
## 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80   
## 2 1 1 3 1 3 1 3 1 1 1 1 3 2 1 2 3 1 1 1   
## 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100   
## 1 3 1 3 2 1 3 1 3 3 1 2 1 1 1 3 1 2 2 1   
## 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120   
## 1 1 3 1 1 1 1 3 1 1 3 1 1 1 3 1 2 1 1 2   
## 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140   
## 3 1 1 2 1 3 2 1 1 1 3 2 1 3 2 2 1 1 2 2   
## 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160   
## 3 3 3 1 3 2 1 1 3 3 2 1 1 1 1 1 1 2 2 1   
## 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180   
## 2 1 3 1 1 1 2 1 3 3 3 1 3 3 1 3 2 1 1 1   
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200   
## 1 3 1 1 3 2 3 3 2 1 1 2 2 1 3 1 3 2 1 3   
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220   
## 1 3 1 2 3 2 1 1 1 1 2 1 1 3 3 1 1 2 1 3   
## 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240   
## 1 1 1 2 1 1 1 3 1 1 1 1 3 3 3 1 1 1 3 3   
## 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260   
## 1 3 3 2 3 3 1 3 3 1 3 2 3 2 3 3 1 2 1 3   
## 261 262 263 264 265 266 267 268 269 270 271 272   
## 1 1 2 3 3 2 2 3 1 1 1 3   
##   
## Within cluster sum of squares by cluster:  
## [1] 471081415 33245844 96006471  
## (between\_SS / total\_SS = 90.8 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

***## Here Cluster 1 formed is best followed by 3. Cluster 2 performs least.***

***## Visualizing Clusters***

fviz\_cluster(clusters, data =as.data.frame(store\_df),  
 palette = c("red", "brown", "blue"),   
 )



***## The clustering algorithm works fairly well. The three clusters formed can be clearly seen.***

#Sorting Total Sales month-wise  
transaction\_data %>%  
 group\_by(MONTH) %>%  
 summarize(tot\_sales=sum(TOT\_SALES))

## # A tibble: 12 x 2  
## MONTH tot\_sales  
## \* <dbl> <dbl>  
## 1 1 162642.  
## 2 2 150665   
## 3 3 166265.  
## 4 4 159845.  
## 5 5 156718.  
## 6 6 160539.  
## 7 7 165275.  
## 8 8 158081.  
## 9 9 160522   
## 10 10 164416.  
## 11 11 160234.  
## 12 12 167913.

***##It can be seen that maximum sales were in the month of December and also the sales don’t vary much month-wise.***

data=transaction\_data %>%  
 group\_by(DATE) %>%  
 summarize(tot\_sales=sum(TOT\_SALES)) %>%  
 mutate(month=month(DATE))

head(data)

## # A tibble: 6 x 3  
## DATE tot\_sales month  
## <date> <dbl> <dbl>  
## 1 2018-07-01 5372. 7  
## 2 2018-07-02 5315. 7  
## 3 2018-07-03 5322. 7  
## 4 2018-07-04 5310. 7  
## 5 2018-07-05 5081. 7  
## 6 2018-07-06 5605. 7

dim(data)

## [1] 364 3

***##There are only 364 rows, meaning only 364 dates which indicates a missing date. So, we create a sequence of dates from 1 Jul 2018 to 30 Jun 2019 and use this to find the missing date.***

date=data$DATE

date\_range=seq(min(date), max(date), by = 1)

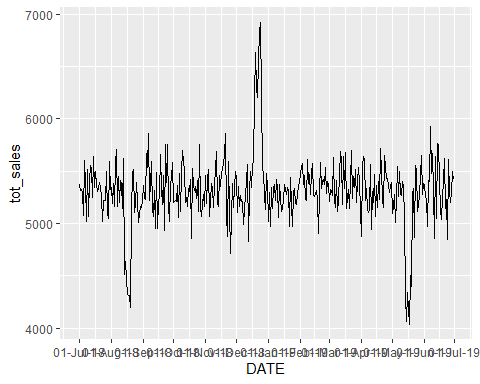
date\_range[!date\_range %in% date]

## "2018-12-25"

***##******It can be observed that there were no transactions on 25th December 2018. It may be that the stores would be closed due to Christmas and Christmas may be the reason that the sales were highest in the month of December.***

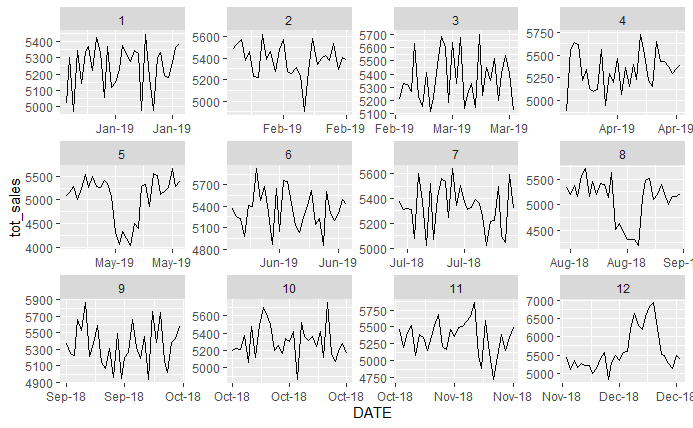
***##Visualizing the total sales***

library(readr)   
library(lubridate)  
library(scales)  
library(ggplot2)  
ggplot(data, aes(x =DATE, y =tot\_sales))+geom\_line()+  
 scale\_x\_date(labels=date\_format("%d-%b-%y"),breaks = ("1 month"))



***##TOTAL SALES MONTH WISE***

ggplot(data, aes(x =DATE, y =tot\_sales))+geom\_line()+  
 scale\_x\_date(labels=date\_format("%b-%y"),breaks=("15 days"))+  
 facet\_wrap(~month,scales = "free")



## Decision tree

#In order to obtain a decision tree, we added a column “level” which would facilitate the binary classification of data along the parameter “total sales”. Total sales greater than 7 are classified as yes meaning it contributed highly in the sales of product and total sales less than 7 implying less contribution comparatively. 7.304 being the mean of the total sales column we chose the reference value to be 7.

level=ifelse(transaction\_data$TOT\_SALES<=7,"No","Yes")  
dec\_data=cbind(transaction\_data[,-c(1,3,4,5,6,8)],level)  
head(dec\_data)

## STORE\_NBR PROD\_QTY MONTH PACK\_SIZE level  
## 1 1 2 10 175 No  
## 2 1 3 5 175 No  
## 3 1 2 5 170 No  
## 4 2 5 8 175 Yes  
## 5 2 3 8 150 Yes  
## 6 4 1 5 300 No

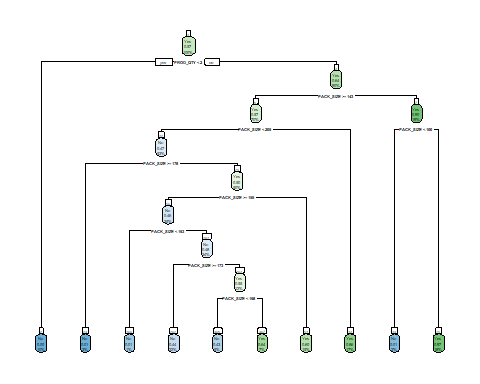
***# We divide 80% data as train data and 20% data as test data.***

library(caTools)  
set.seed(987)  
sample=sample.split(dec\_data, SplitRatio=0.8)  
train=subset(dec\_data, sample==TRUE)  
test=subset(dec\_data,sample==FALSE)  
library(rpart)  
tree=rpart(level~.,data=train,method = "class")  
tree

## n= 211867   
##   
## node), split, n, loss, yval, (yprob)  
## \* denotes terminal node  
##   
## 1) root 211867 90226 Yes (0.425861507 0.574138493)   
## 2) PROD\_QTY< 1.5 22027 0 No (1.000000000 0.000000000) \*  
## 3) PROD\_QTY>=1.5 189840 68199 Yes (0.359244627 0.640755373)   
## 6) PACK\_SIZE>=142.5 148967 64029 Yes (0.429820027 0.570179973)   
## 12) PACK\_SIZE< 205 111999 53149 No (0.525451120 0.474548880)   
## 24) PACK\_SIZE>=177.5 6201 32 No (0.994839542 0.005160458) \*  
## 25) PACK\_SIZE< 177.5 105798 52681 Yes (0.497939470 0.502060530)   
## 50) PACK\_SIZE>=155 74794 34625 No (0.537061796 0.462938204)   
## 100) PACK\_SIZE< 162.5 2057 13 No (0.993680117 0.006319883) \*  
## 101) PACK\_SIZE>=162.5 72737 34612 No (0.524148645 0.475851355)   
## 202) PACK\_SIZE>=172.5 47485 20759 No (0.562830367 0.437169633) \*  
## 203) PACK\_SIZE< 172.5 25252 11399 Yes (0.451409789 0.548590211)   
## 406) PACK\_SIZE< 167.5 10894 4645 No (0.573618506 0.426381494) \*  
## 407) PACK\_SIZE>=167.5 14358 5150 Yes (0.358685054 0.641314946) \*  
## 51) PACK\_SIZE< 155 31004 12512 Yes (0.403560831 0.596439169) \*  
## 13) PACK\_SIZE>=205 36968 5179 Yes (0.140094135 0.859905865) \*  
## 7) PACK\_SIZE< 142.5 40873 4170 Yes (0.102023341 0.897976659)   
## 14) PACK\_SIZE< 100 3189 19 No (0.994042019 0.005957981) \*  
## 15) PACK\_SIZE>=100 37684 1000 Yes (0.026536461 0.973463539) \*

**##It comes out that the product quantity and pack size being the major split node.**

# Visualize the decision tree with rpart.plot  
library(rpart.plot)  
rpart.plot(tree,nn=TRUE)



Prediction1=predict(tree,newdata=test,type = 'class')

# Confusion matrix  
library(caret)  
library(e1071)  
confusionMatrix(as.factor(Prediction1),as.factor(test$level))

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction No Yes  
## No 16684 6402  
## Yes 5877 24004  
##   
## Accuracy : 0.7682   
## 95% CI : (0.7646, 0.7718)  
## No Information Rate : 0.5741   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.5274   
##   
## Mcnemar's Test P-Value : 2.259e-06   
##   
## Sensitivity : 0.7395   
## Specificity : 0.7894   
## Pos Pred Value : 0.7227   
## Neg Pred Value : 0.8033   
## Prevalence : 0.4259   
## Detection Rate : 0.3150   
## Detection Prevalence : 0.4359   
## Balanced Accuracy : 0.7645   
##   
## 'Positive' Class : No   
##

***##We find a decision tree with 76.82% accuracy, 73.95% Sensitivity and 78.9% specificity.***

CONCLUSIONS:

Some of the major insights drawn from analysis were:

1) Older families were contributing most towards total sales followed by young families and Retirees. New families contributing least, its due to the fact that new-families themselves being least customers in number.

2) The product “Chips” is most purchased product and especially of “Kettle” brand followed by “RRD” brand.

3)Pack size of 175g is most preferred by customers.

4)Maximum sales were in the month of December and also the sales don’t vary much month-wise.

5) It can be observed that there were no transactions on 25th December 2018. It may be that the stores would be closed due to Christmas and Christmas may be the reason that the sales were highest in the month of December. It the only day when there were no transactions.

6)The stores with store number 226, 88, 93, 165 and 237 contributes the most***.***

7) While creating a Decision tree it comes out that the product quantity and pack size being the major split node.

8) We find a decision tree with 76.82% accuracy, 73.95% Sensitivity and 78.9% specificity.

9) Appropriate clusters were formed based on stores and products.