Imarticus project

Team

17/11/2020

cd <- read.csv("D:/R Markdown/1cereals\_data.csv", stringsAsFactors=TRUE)  
wt\_cal<-cd[,c(1,2,4,14)]  
wt\_cal

## name mfr calories weight  
## 1 100% Bran N 70 1.00  
## 2 100% Natural Bran Q 120 1.00  
## 3 All-Bran K 70 1.00  
## 4 All-Bran with Extra Fiber K 50 1.00  
## 5 Almond Delight R 110 1.00  
## 6 Apple Cinnamon Cheerios G 110 1.00  
## 7 Apple Jacks K 110 1.00  
## 8 Basic 4 G 130 1.33  
## 9 Bran Chex R 90 1.00  
## 10 Bran Flakes P 90 1.00  
## 11 Cap'n'Crunch Q 120 1.00  
## 12 Cheerios G 110 1.00  
## 13 Cinnamon Toast Crunch G 120 1.00  
## 14 Clusters G 110 1.00  
## 15 Cocoa Puffs G 110 1.00  
## 16 Corn Chex R 110 1.00  
## 17 Corn Flakes K 100 1.00  
## 18 Corn Pops K 110 1.00  
## 19 Count Chocula G 110 1.00  
## 20 Cracklin' Oat Bran K 110 1.00  
## 21 Cream of Wheat (Quick) N 100 1.00  
## 22 Crispix K 110 1.00  
## 23 Crispy Wheat & Raisins G 100 1.00  
## 24 Double Chex R 100 1.00  
## 25 Froot Loops K 110 1.00  
## 26 Frosted Flakes K 110 1.00  
## 27 Frosted Mini-Wheats K 100 1.00  
## 28 Fruit & Fibre Dates, Walnuts, and Oats P 120 1.25  
## 29 Fruitful Bran K 120 1.33  
## 30 Fruity Pebbles P 110 1.00  
## 31 Golden Crisp P 100 1.00  
## 32 Golden Grahams G 110 1.00  
## 33 Grape Nuts Flakes P 100 1.00  
## 34 Grape-Nuts P 110 1.00  
## 35 Great Grains Pecan P 120 1.00  
## 36 Honey Graham Ohs Q 120 1.00  
## 37 Honey Nut Cheerios G 110 1.00  
## 38 Honey-comb P 110 1.00  
## 39 Just Right Crunchy Nuggets K 110 1.00  
## 40 Just Right Fruit & Nut K 140 1.30  
## 41 Kix G 110 1.00  
## 42 Life Q 100 1.00  
## 43 Lucky Charms G 110 1.00  
## 44 Maypo A 100 1.00  
## 45 Muesli Raisins, Dates, & Almonds R 150 1.00  
## 46 Muesli Raisins, Peaches, & Pecans R 150 1.00  
## 47 Mueslix Crispy Blend K 160 1.50  
## 48 Multi-Grain Cheerios G 100 1.00  
## 49 Nut&Honey Crunch K 120 1.00  
## 50 Nutri-Grain Almond-Raisin K 140 1.33  
## 51 Nutri-grain Wheat K 90 1.00  
## 52 Oatmeal Raisin Crisp G 130 1.25  
## 53 Post Nat. Raisin Bran P 120 1.33  
## 54 Product 19 K 100 1.00  
## 55 Puffed Rice Q 50 0.50  
## 56 Puffed Wheat Q 50 0.50  
## 57 Quaker Oat Squares Q 100 1.00  
## 58 Quaker Oatmeal Q 100 1.00  
## 59 Raisin Bran K 120 1.33  
## 60 Raisin Nut Bran G 100 1.00  
## 61 Raisin Squares K 90 1.00  
## 62 Rice Chex R 110 1.00  
## 63 Rice Krispies K 110 1.00  
## 64 Shredded Wheat N 80 0.83  
## 65 Shredded Wheat 'n'Bran N 90 1.00  
## 66 Shredded Wheat spoon size N 90 1.00  
## 67 Smacks K 110 1.00  
## 68 Special K K 110 1.00  
## 69 Strawberry Fruit Wheats N 90 1.00  
## 70 Total Corn Flakes G 110 1.00  
## 71 Total Raisin Bran G 140 1.50  
## 72 Total Whole Grain G 100 1.00  
## 73 Triples G 110 1.00  
## 74 Trix G 110 1.00  
## 75 Wheat Chex R 100 1.00  
## 76 Wheaties G 100 1.00  
## 77 Wheaties Honey Gold G 110 1.00

View(wt\_cal)

#Build corpus  
library(tm)

## Loading required package: NLP

corpus<-iconv(wt\_cal$name, to='UTF-8', sub = "byte")  
corpus<-Corpus(VectorSource(corpus))  
inspect(corpus[1:5])

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 5  
##   
## [1] 100% Bran 100% Natural Bran   
## [3] All-Bran All-Bran with Extra Fiber  
## [5] Almond Delight

#clean tm\_map(0)  
corpus<-tm\_map(corpus,tolower)

## Warning in tm\_map.SimpleCorpus(corpus, tolower): transformation drops documents

inspect(corpus)

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 77  
##   
## [1] 100% bran   
## [2] 100% natural bran   
## [3] all-bran   
## [4] all-bran with extra fiber   
## [5] almond delight   
## [6] apple cinnamon cheerios   
## [7] apple jacks   
## [8] basic 4   
## [9] bran chex   
## [10] bran flakes   
## [11] cap'n'crunch   
## [12] cheerios   
## [13] cinnamon toast crunch   
## [14] clusters   
## [15] cocoa puffs   
## [16] corn chex   
## [17] corn flakes   
## [18] corn pops   
## [19] count chocula   
## [20] cracklin' oat bran   
## [21] cream of wheat (quick)   
## [22] crispix   
## [23] crispy wheat & raisins   
## [24] double chex   
## [25] froot loops   
## [26] frosted flakes   
## [27] frosted mini-wheats   
## [28] fruit & fibre dates, walnuts, and oats  
## [29] fruitful bran   
## [30] fruity pebbles   
## [31] golden crisp   
## [32] golden grahams   
## [33] grape nuts flakes   
## [34] grape-nuts   
## [35] great grains pecan   
## [36] honey graham ohs   
## [37] honey nut cheerios   
## [38] honey-comb   
## [39] just right crunchy nuggets   
## [40] just right fruit & nut   
## [41] kix   
## [42] life   
## [43] lucky charms   
## [44] maypo   
## [45] muesli raisins, dates, & almonds   
## [46] muesli raisins, peaches, & pecans   
## [47] mueslix crispy blend   
## [48] multi-grain cheerios   
## [49] nut&honey crunch   
## [50] nutri-grain almond-raisin   
## [51] nutri-grain wheat   
## [52] oatmeal raisin crisp   
## [53] post nat. raisin bran   
## [54] product 19   
## [55] puffed rice   
## [56] puffed wheat   
## [57] quaker oat squares   
## [58] quaker oatmeal   
## [59] raisin bran   
## [60] raisin nut bran   
## [61] raisin squares   
## [62] rice chex   
## [63] rice krispies   
## [64] shredded wheat   
## [65] shredded wheat 'n'bran   
## [66] shredded wheat spoon size   
## [67] smacks   
## [68] special k   
## [69] strawberry fruit wheats   
## [70] total corn flakes   
## [71] total raisin bran   
## [72] total whole grain   
## [73] triples   
## [74] trix   
## [75] wheat chex   
## [76] wheaties   
## [77] wheaties honey gold

corpus<-tm\_map(corpus,removeNumbers)

## Warning in tm\_map.SimpleCorpus(corpus, removeNumbers): transformation drops  
## documents

inspect(corpus)

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 77  
##   
## [1] % bran   
## [2] % natural bran   
## [3] all-bran   
## [4] all-bran with extra fiber   
## [5] almond delight   
## [6] apple cinnamon cheerios   
## [7] apple jacks   
## [8] basic   
## [9] bran chex   
## [10] bran flakes   
## [11] cap'n'crunch   
## [12] cheerios   
## [13] cinnamon toast crunch   
## [14] clusters   
## [15] cocoa puffs   
## [16] corn chex   
## [17] corn flakes   
## [18] corn pops   
## [19] count chocula   
## [20] cracklin' oat bran   
## [21] cream of wheat (quick)   
## [22] crispix   
## [23] crispy wheat & raisins   
## [24] double chex   
## [25] froot loops   
## [26] frosted flakes   
## [27] frosted mini-wheats   
## [28] fruit & fibre dates, walnuts, and oats  
## [29] fruitful bran   
## [30] fruity pebbles   
## [31] golden crisp   
## [32] golden grahams   
## [33] grape nuts flakes   
## [34] grape-nuts   
## [35] great grains pecan   
## [36] honey graham ohs   
## [37] honey nut cheerios   
## [38] honey-comb   
## [39] just right crunchy nuggets   
## [40] just right fruit & nut   
## [41] kix   
## [42] life   
## [43] lucky charms   
## [44] maypo   
## [45] muesli raisins, dates, & almonds   
## [46] muesli raisins, peaches, & pecans   
## [47] mueslix crispy blend   
## [48] multi-grain cheerios   
## [49] nut&honey crunch   
## [50] nutri-grain almond-raisin   
## [51] nutri-grain wheat   
## [52] oatmeal raisin crisp   
## [53] post nat. raisin bran   
## [54] product   
## [55] puffed rice   
## [56] puffed wheat   
## [57] quaker oat squares   
## [58] quaker oatmeal   
## [59] raisin bran   
## [60] raisin nut bran   
## [61] raisin squares   
## [62] rice chex   
## [63] rice krispies   
## [64] shredded wheat   
## [65] shredded wheat 'n'bran   
## [66] shredded wheat spoon size   
## [67] smacks   
## [68] special k   
## [69] strawberry fruit wheats   
## [70] total corn flakes   
## [71] total raisin bran   
## [72] total whole grain   
## [73] triples   
## [74] trix   
## [75] wheat chex   
## [76] wheaties   
## [77] wheaties honey gold

cleanset<-tm\_map(corpus,removeWords,stopwords("en"))

## Warning in tm\_map.SimpleCorpus(corpus, removeWords, stopwords("en")):  
## transformation drops documents

cleanset<-tm\_map(corpus,removeWords,c("total","100%"))

## Warning in tm\_map.SimpleCorpus(corpus, removeWords, c("total", "100%")):  
## transformation drops documents

inspect(cleanset[1:5])

## <<SimpleCorpus>>  
## Metadata: corpus specific: 1, document level (indexed): 0  
## Content: documents: 5  
##   
## [1] % bran % natural bran   
## [3] all-bran all-bran with extra fiber  
## [5] almond delight

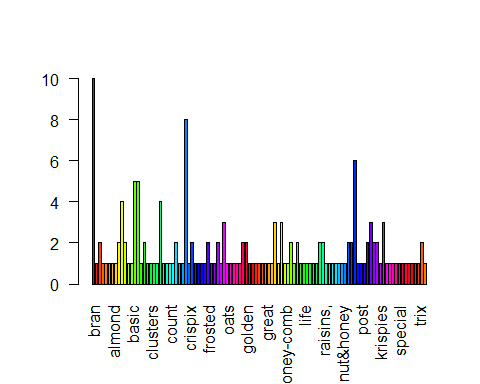
#term document matrix  
tdm<-TermDocumentMatrix(cleanset)  
tdm

## <<TermDocumentMatrix (terms: 105, documents: 77)>>  
## Non-/sparse entries: 170/7915  
## Sparsity : 98%  
## Maximal term length: 13  
## Weighting : term frequency (tf)

tdm<-as.matrix(tdm)  
tdm[1:10,1:20]

## Docs  
## Terms 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20  
## bran 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 1  
## natural 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## all-bran 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## extra 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## fiber 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## with 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## almond 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## delight 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
## apple 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0  
## cheerios 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0

w<-rowSums(tdm)  
barplot(w,las=2,col=rainbow(50))



##Word cloud  
library(wordcloud)

## Loading required package: RColorBrewer

w<-sort(rowSums(tdm), decreasing = T)  
set.seed(222)  
wordcloud(words = names(w),freq = w,max.words = 1500, random.order = FALSE,colors = brewer.pal(10,"Dark2"),scale =c(4,.3))

## Warning in brewer.pal(10, "Dark2"): n too large, allowed maximum for palette Dark2 is 8  
## Returning the palette you asked for with that many colors



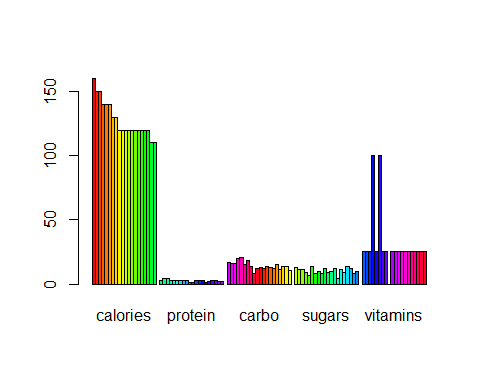
#install.packages("tidyverse")  
library("tidyverse")

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2 v purrr 0.3.4  
## v tibble 3.0.4 v dplyr 1.0.2  
## v tidyr 1.1.2 v stringr 1.4.0  
## v readr 1.4.0 v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x ggplot2::annotate() masks NLP::annotate()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

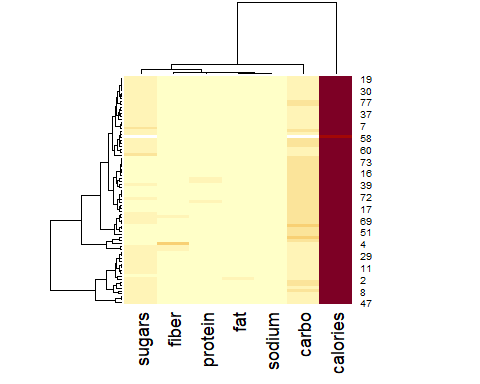
cd <-read.csv("D:/R Markdown/1cereals\_data.csv", stringsAsFactors=TRUE)  
view(cd)  
#selecting data for plot  
Imp\_nut<-cd[,c(4,5,9,10,12)]  
view(Imp\_nut)  
Imp\_nut\_top<-head(Imp\_nut[order(Imp\_nut$calories, decreasing=TRUE), ], 20)  
view(Imp\_nut\_top)  
Imp\_nut\_top<-as.matrix(Imp\_nut\_top)  
mode(Imp\_nut\_top)="numeric"  
#Imp\_nut\_top<-  
barplot(Imp\_nut\_top,beside = TRUE,col = rainbow(50))



cd\_hitmap <- read.csv("D:/R Markdown/1cereals\_data.csv", stringsAsFactors=TRUE)  
cd\_hitmap$sodium<-cd\_hitmap$sodium/100  
cd\_hitmap<-data.matrix(cd\_hitmap)  
View(cd\_hitmap)  
  
  
cd\_hitmap<-cd\_hitmap[,4:10]  
cd\_hitmap

## calories protein fat sodium fiber carbo sugars  
## [1,] 70 4 1 1.30 10.0 5.0 6  
## [2,] 120 3 5 0.15 2.0 8.0 8  
## [3,] 70 4 1 2.60 9.0 7.0 5  
## [4,] 50 4 0 1.40 14.0 8.0 0  
## [5,] 110 2 2 2.00 1.0 14.0 8  
## [6,] 110 2 2 1.80 1.5 10.5 10  
## [7,] 110 2 0 1.25 1.0 11.0 14  
## [8,] 130 3 2 2.10 2.0 18.0 8  
## [9,] 90 2 1 2.00 4.0 15.0 6  
## [10,] 90 3 0 2.10 5.0 13.0 5  
## [11,] 120 1 2 2.20 0.0 12.0 12  
## [12,] 110 6 2 2.90 2.0 17.0 1  
## [13,] 120 1 3 2.10 0.0 13.0 9  
## [14,] 110 3 2 1.40 2.0 13.0 7  
## [15,] 110 1 1 1.80 0.0 12.0 13  
## [16,] 110 2 0 2.80 0.0 22.0 3  
## [17,] 100 2 0 2.90 1.0 21.0 2  
## [18,] 110 1 0 0.90 1.0 13.0 12  
## [19,] 110 1 1 1.80 0.0 12.0 13  
## [20,] 110 3 3 1.40 4.0 10.0 7  
## [21,] 100 3 0 0.80 1.0 21.0 0  
## [22,] 110 2 0 2.20 1.0 21.0 3  
## [23,] 100 2 1 1.40 2.0 11.0 10  
## [24,] 100 2 0 1.90 1.0 18.0 5  
## [25,] 110 2 1 1.25 1.0 11.0 13  
## [26,] 110 1 0 2.00 1.0 14.0 11  
## [27,] 100 3 0 0.00 3.0 14.0 7  
## [28,] 120 3 2 1.60 5.0 12.0 10  
## [29,] 120 3 0 2.40 5.0 14.0 12  
## [30,] 110 1 1 1.35 0.0 13.0 12  
## [31,] 100 2 0 0.45 0.0 11.0 15  
## [32,] 110 1 1 2.80 0.0 15.0 9  
## [33,] 100 3 1 1.40 3.0 15.0 5  
## [34,] 110 3 0 1.70 3.0 17.0 3  
## [35,] 120 3 3 0.75 3.0 13.0 4  
## [36,] 120 1 2 2.20 1.0 12.0 11  
## [37,] 110 3 1 2.50 1.5 11.5 10  
## [38,] 110 1 0 1.80 0.0 14.0 11  
## [39,] 110 2 1 1.70 1.0 17.0 6  
## [40,] 140 3 1 1.70 2.0 20.0 9  
## [41,] 110 2 1 2.60 0.0 21.0 3  
## [42,] 100 4 2 1.50 2.0 12.0 6  
## [43,] 110 2 1 1.80 0.0 12.0 12  
## [44,] 100 4 1 0.00 0.0 16.0 3  
## [45,] 150 4 3 0.95 3.0 16.0 11  
## [46,] 150 4 3 1.50 3.0 16.0 11  
## [47,] 160 3 2 1.50 3.0 17.0 13  
## [48,] 100 2 1 2.20 2.0 15.0 6  
## [49,] 120 2 1 1.90 0.0 15.0 9  
## [50,] 140 3 2 2.20 3.0 21.0 7  
## [51,] 90 3 0 1.70 3.0 18.0 2  
## [52,] 130 3 2 1.70 1.5 13.5 10  
## [53,] 120 3 1 2.00 6.0 11.0 14  
## [54,] 100 3 0 3.20 1.0 20.0 3  
## [55,] 50 1 0 0.00 0.0 13.0 0  
## [56,] 50 2 0 0.00 1.0 10.0 0  
## [57,] 100 4 1 1.35 2.0 14.0 6  
## [58,] 100 5 2 0.00 2.7 NA NA  
## [59,] 120 3 1 2.10 5.0 14.0 12  
## [60,] 100 3 2 1.40 2.5 10.5 8  
## [61,] 90 2 0 0.00 2.0 15.0 6  
## [62,] 110 1 0 2.40 0.0 23.0 2  
## [63,] 110 2 0 2.90 0.0 22.0 3  
## [64,] 80 2 0 0.00 3.0 16.0 0  
## [65,] 90 3 0 0.00 4.0 19.0 0  
## [66,] 90 3 0 0.00 3.0 20.0 0  
## [67,] 110 2 1 0.70 1.0 9.0 15  
## [68,] 110 6 0 2.30 1.0 16.0 3  
## [69,] 90 2 0 0.15 3.0 15.0 5  
## [70,] 110 2 1 2.00 0.0 21.0 3  
## [71,] 140 3 1 1.90 4.0 15.0 14  
## [72,] 100 3 1 2.00 3.0 16.0 3  
## [73,] 110 2 1 2.50 0.0 21.0 3  
## [74,] 110 1 1 1.40 0.0 13.0 12  
## [75,] 100 3 1 2.30 3.0 17.0 3  
## [76,] 100 3 1 2.00 3.0 17.0 3  
## [77,] 110 2 1 2.00 1.0 16.0 8

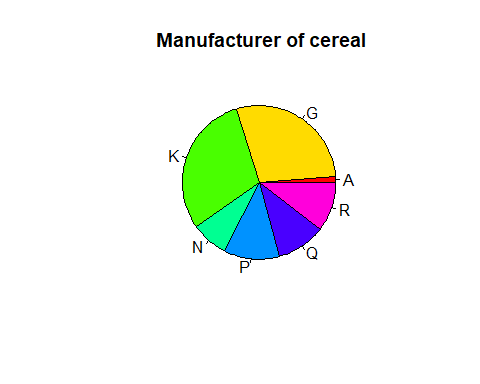
heatmap(cd\_hitmap)



data1<- read.csv("D:/R Markdown/1cereals\_data.csv", stringsAsFactors=TRUE)  
library(RColorBrewer)  
data1$mfrFlag <- as.numeric(!is.na(data1$calories))  
pichartdata<-tapply(data1$mfrFlag , data1$mfr,sum)  
pichartdata

## A G K N P Q R   
## 1 22 23 6 9 8 8

pichartdata<-as.matrix(pichartdata)  
pie(pichartdata[,1],main = " Manufacturer of cereal",col = rainbow(length(pichartdata)))



## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.