

# Sistema de Recomendação de Produtos

Marlon Moreira

21/08/2021

## Sistema de Recomendação de Produtos

O objetivo desse projeto é criar um sistema que possa recomendar produtos que são comprados em conjunto com muita frequência. Para isso vou usar o dataset Online-Retail que está disponível na UCI Machine Learning Repository. Esse conjunto dados contém todas as transações ocorridas entre 01/12/2010 e 09/12/2011 de uma empresa de varejo online do Reino Unido.

## Carregando os pacotes e visualizando os dados

```
# carregando os pacotes
library(arules)

## Loading required package: Matrix

##
## Attaching package: 'arules'

## The following objects are masked from 'package:base':
##
##      abbreviate, write

library(arulesViz)
library(tidyverse)

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

## v ggplot2 3.3.3      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 tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x dplyr::recode() masks arules::recode()
## x tidyr::unpack() masks Matrix::unpack()
```

```
library(readxl)
library(knitr)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:arules':
##
## intersect, setdiff, union

## The following objects are masked from 'package:base':
##
## date, intersect, setdiff, union
```

```
library(plyr)
```

```
## -----

## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)

## -----

##
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':
##
## arrange, count, desc, failwith, id, mutate, rename, summarise,
## summarize

## The following object is masked from 'package:purrr':
##
## compact
```

```
library(dplyr)
library(RColorBrewer)

# carregando os dados
data = read_excel('Online Retail.xlsx')
View(data)
```

## Análise Exploratória

Os dados possui outliers e alguns valores faltantes. Há muitos valores missing na coluna do id do cliente. Além disso, as compras em sua maioria é de produtos com preço unitário baixo e em pouca quantidade.

```
# fazendo um resumo inicial dos dados
summary(data[,c(4,6)])
```

```
##      Quantity      UnitPrice
## Min.   :-80995.00 Min.   :-11062.06
## 1st Qu.:   1.00 1st Qu.:   1.25
## Median :   3.00 Median :   2.08
## Mean   :   9.55 Mean   :   4.61
## 3rd Qu.:  10.00 3rd Qu.:   4.13
## Max.   : 80995.00 Max.   : 38970.00
```

```
str(data)
```

```
## tibble [541,909 x 8] (S3: tbl_df/tbl/data.frame)
## $ InvoiceNo : chr [1:541909] "536365" "536365" "536365" "536365" ...
## $ StockCode : chr [1:541909] "85123A" "71053" "84406B" "84029G" ...
## $ Description: chr [1:541909] "WHITE HANGING HEART T-LIGHT HOLDER" "WHITE METAL LANTERN" "CREAM CUP" ...
## $ Quantity : num [1:541909] 6 6 8 6 6 2 6 6 6 32 ...
## $ InvoiceDate: POSIXct[1:541909], format: "2010-12-01 08:26:00" "2010-12-01 08:26:00" ...
## $ UnitPrice : num [1:541909] 2.55 3.39 2.75 3.39 3.39 7.65 4.25 1.85 1.85 1.69 ...
## $ CustomerID: num [1:541909] 17850 17850 17850 17850 17850 ...
## $ Country : chr [1:541909] "United Kingdom" "United Kingdom" "United Kingdom" "United Kingdom" .
```

```
# No resumo as colunas preço unitário e quantidade tem
# como registro mínimo valor negativo
# investigando pra ver se há mais
neg_quant = data %>%
  filter(Quantity < 0)

neg_price = data %>%
  filter(UnitPrice < 0)

View(neg_price)
View(neg_quant)

# Como se trata de promoções e algo do tipo vou retirar esses
# do dataset
data2 = data[!data$Quantity < 0 & !data$UnitPrice < 0,]

summary(data2[,c(4,6)])
```

```
##      Quantity      UnitPrice
## Min.   :   1.00 Min.   :   0.000
## 1st Qu.:   1.00 1st Qu.:   1.250
## Median :   3.00 Median :   2.080
## Mean   :  10.66 Mean   :   3.899
## 3rd Qu.:  10.00 3rd Qu.:   4.130
## Max.   : 80995.00 Max.   :13541.330
```

```
# Vi também que há muitos valores nulos
sum(is.na(data2))
```

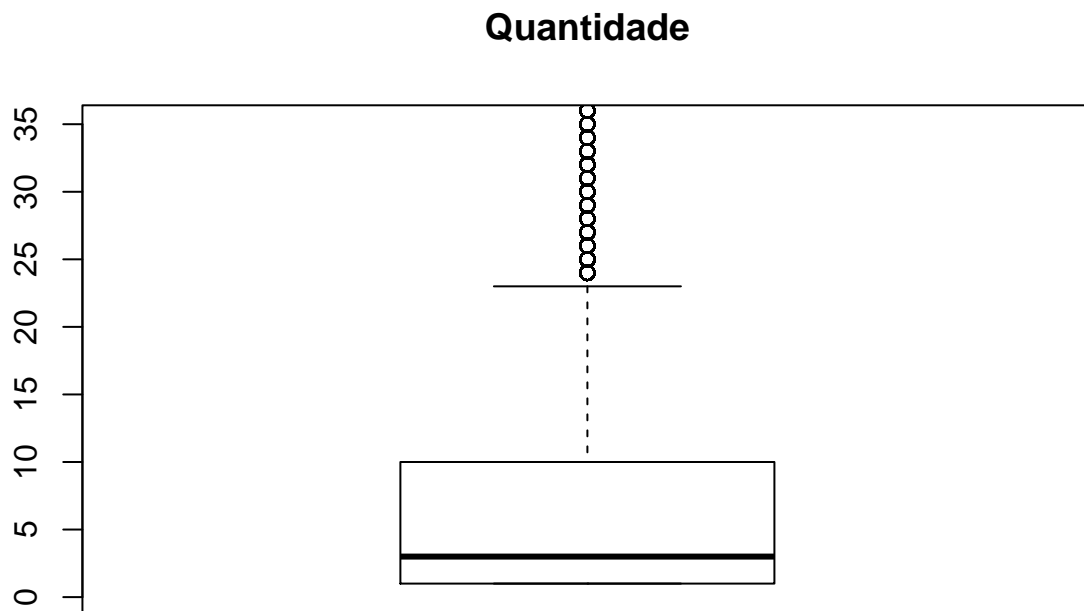
```
## [1] 133951
```

```
for (i in colnames(data2)) {  
  print(sum(is.na(data2[i])))  
}
```

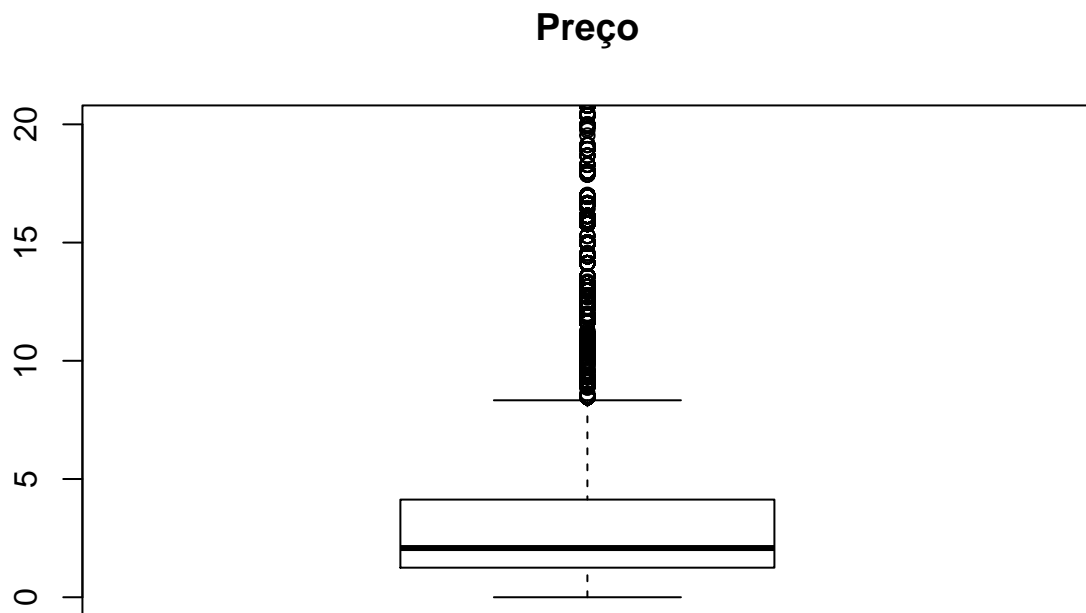
```
## [1] 0  
## [1] 0  
## [1] 592  
## [1] 0  
## [1] 0  
## [1] 0  
## [1] 133359  
## [1] 0
```

```
# Em sua maioria os valores nulos está no id do cliente
```

```
boxplot(data2$Quantity, main = 'Quantidade',ylim=c(0,35))
```



```
boxplot(data2$UnitPrice, main = 'Preço',ylim=c(0,20))
```



```
# como há alguns outliers vou fazer um filtro
quantile(data2$Quantity, seq(from = 0,to = 1,by = .10))
```

```
##      0%      10%      20%      30%      40%      50%      60%      70%      80%      90%     100%
##       1        1        1        2        2        3        6        8       12       24    80995
```

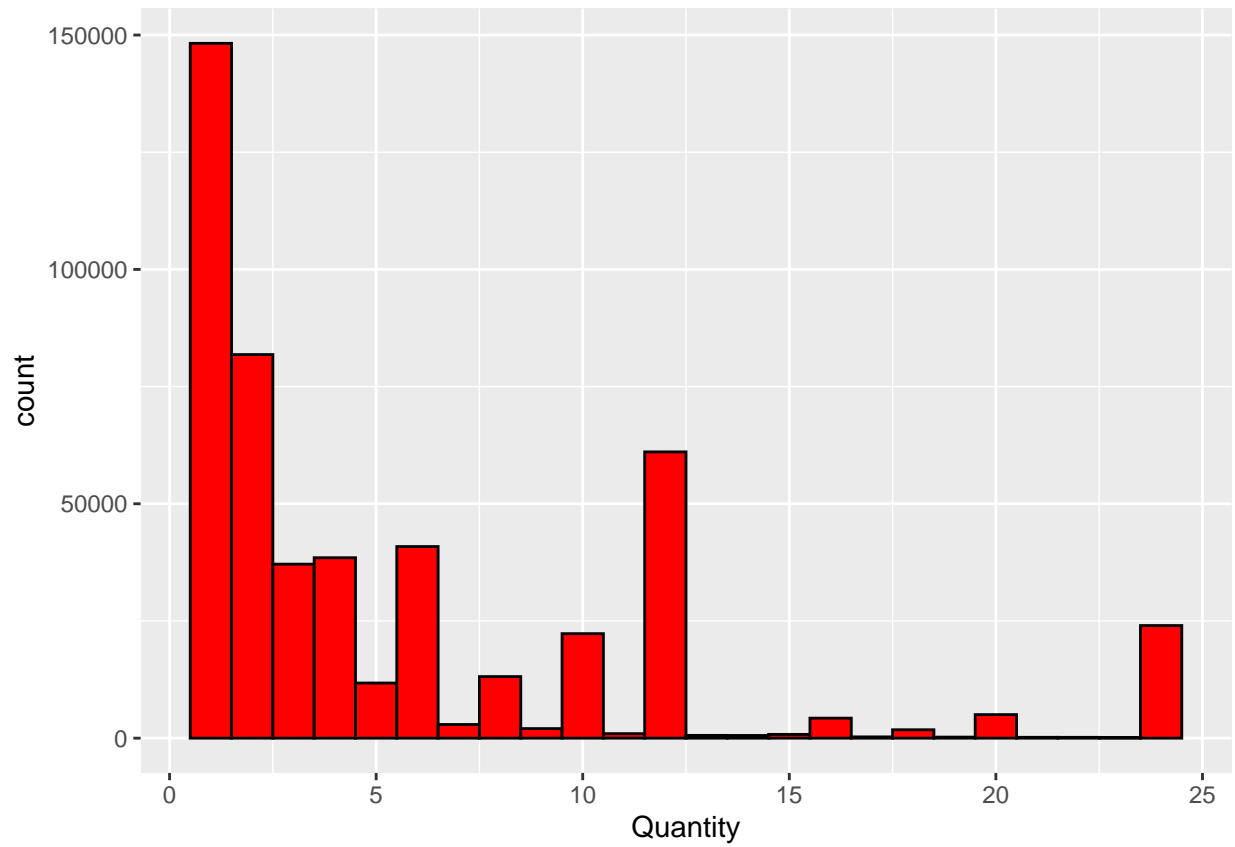
```
quantile(data2$UnitPrice, seq(from = 0,to = 1,by = .10))
```

```
##          0%          10%          20%          30%          40%          50%          60%          70%
##         0.00         0.65         0.85         1.25         1.65         2.08         2.55         3.75
##          80%          90%         100%
##         4.95         7.95      13541.33
```

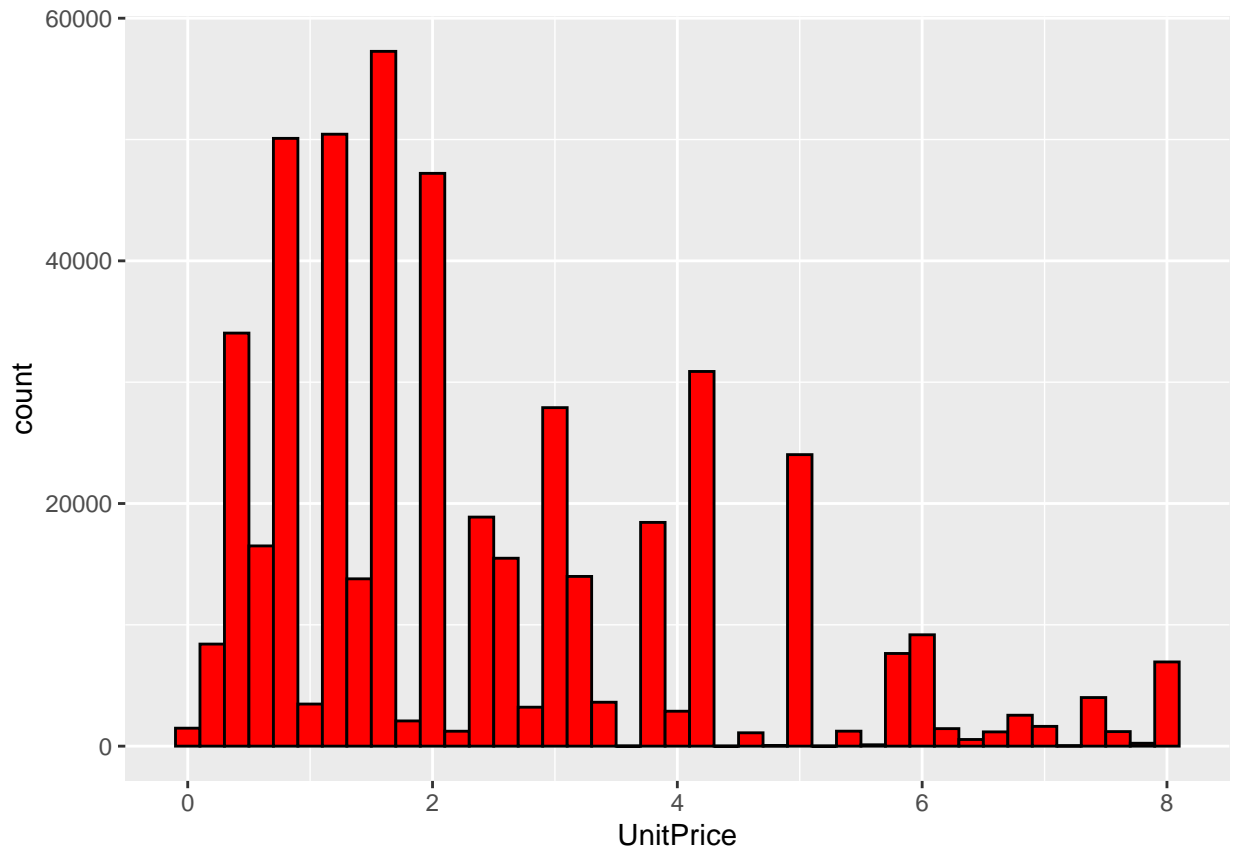
```
# pegando somente os dados sem os outliers
quant = data2 %>%
  filter(Quantity < 25)

price = data2 %>%
  filter(UnitPrice < 8)

ggplot(quant,aes(Quantity)) +
  geom_histogram(binwidth = 1,col = 'black', fill= 'red')
```



```
ggplot(price,aes(UnitPrice)) +  
  geom_histogram(binwidth = 0.2,col = 'black', fill= 'red')
```



### ## Pré-Processamento

Nessa etapa eu apliquei algumas transformações no dataset para que ele fique pronto para o algoritmo. Os dados precisam estar em um formato de dados de transação para que o APRIORI consiga fazer as associações.

```
# A quantidade mais comprada (1 e 2) e depois (3,4 e 6) e o valor
# unitário até 2.0
```

```
# Agora irei aplicar algumas transformações no dataset.
# Ele necessita estar em um formato de dados de transação
# Todos os itens que foram comprados em uma invoice fiquem
# na mesma linha. Assim, o algoritmo conseguirá criar as
# regras de associação.
```

```
# Passar algumas variáveis para o tipo de dado correto.
```

```
data2$InvoiceNo = as.numeric(data2$InvoiceNo)
```

## Warning: NAs introduzidos por coerção

```
data2$Country = as.factor(data2$Country)
```

```
# criando essa variável de hora para pegar todas as compras de
# uma invoice.
```

```
data2$date = format(data2$InvoiceDate,"%H:%M:%S")
```

```
# Poderia fazer pelo id do cliente, mas há valores faltantes  
# então optei pelo invoice.
```

```
# esse trecho de código irá usar as duas colunas  
# invoice e date para agrupar os dados e a função anônima  
# irá separar por vírgula todos os produtos comprados na  
# mesma invoice.  
transactiondata = ddply(data2,c("InvoiceNo","date"),  
                        function(df1)paste(df1$Description,  
                                           collapse = ","))
```

```
View(transactiondata)
```

```
# Não é necessário usar as colunas invoiceno e date  
# por isso vou deletá-las  
colnames(transactiondata)[3] = 'produtos'  
transactiondata$InvoiceNo = NULL  
transactiondata$date = NULL  
  
# gravar os dados em um arquivo csv para depois  
# transformá-lo em dados de transação  
write.csv(transactiondata,"transactions.csv", quote = FALSE, row.names = FALSE)  
# Aqui é que deixarei os dados no formato de transação de fato  
transacao = read.transactions('transactions.csv', format = 'basket', sep=',')
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```

```
## Warning in scan(text = 1, what = "character", sep = sep, quote = quote, : EOF  
## within quoted string
```



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

























[illegible]

[illegible]

[illegible]



[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]



[illegible]



[illegible]



[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



```

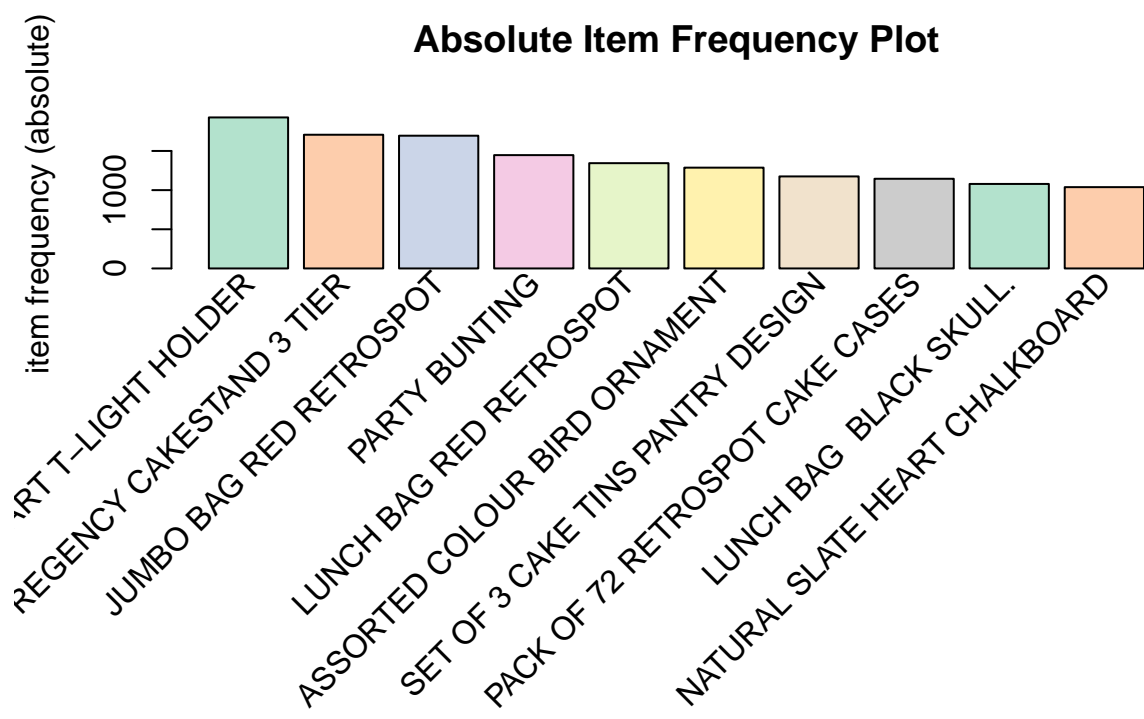
##                                     1344                                     404979
##
## element (itemset/transaction) length distribution:
## sizes
##   0    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
## 592 1978 973 823 804 792 734 674 666 659 600 631 536 519 540 559
##  16   17   18   19   20   21   22   23   24   25   26   27   28   29   30   31
## 531 482 451 490 430 405 326 317 284 248 261 235 225 232 220 170
##  32   33   34   35   36   37   38   39   40   41   42   43   44   45   46   47
## 172 145 149 140 118 120 101 115 99 93 95 94 73 71 73 69
##  48   49   50   51   52   53   54   55   56   57   58   59   60   61   62   63
##  69   64   55   66   46   52   55   52   40   33   44   37   31   34   20   26
##  64   65   66   67   68   69   70   71   72   73   74   75   76   77   78   79
##  27   22   31   26   29   18   22   23   15   17   23   11   16   14   13    9
##  80   81   82   83   84   85   86   87   88   89   90   91   92   93   94   95
##  18   18   15    8    9   15   13   16   11    9    8   12   12    8    7    7
##  96   97   98   99  100  101  102  103  104  105  106  107  108  109  110  111
##   4    8    9    4    8    5    4    5    6    2    3    7    9    4    8    4
## 112 113 114 116 117 118 119 120 121 122 123 124 125 126 127 128
##   2    7    1    4    7    5    1    3    6    4    3    2    5    5    2    1
## 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   1    4    3    5    5    2    4    3    1    1    1    3    7    5    3    3
## 145 146 147 148 150 151 152 153 154 155 156 157 158 159 160 162
##   3    7    2    3    3    3    2    4    7    3    3    5    1    4    4    1
## 163 164 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   2    2    3    5    2    2    3    2    1    2    5    1    1    4    3    2
## 181 182 183 184 185 186 187 189 192 193 194 196 197 198 201 202
##   1    2    1    2    1    1    2    2    1    4    1    3    3    1    1    1
## 204 205 206 207 208 209 212 213 215 217 219 220 224 226 227 228
##   2    1    1    2    3    2    1    1    2    1    3    1    3    3    1    1
## 230 232 234 238 240 241 248 249 250 252 256 257 258 260 261 263
##   2    1    1    2    1    2    1    1    2    1    1    1    1    2    1    1
## 265 266 270 272 274 281 284 285 291 298 301 303 305 312 314 320
##   1    1    1    1    1    1    1    1    1    1    1    1    2    2    1    2
## 321 326 327 329 332 338 339 343 344 348 350 360 365 367 375 391
##   1    1    1    1    1    1    1    1    1    1    1    2    1    1    3    1
## 394 398 400 402 411 419 429 431 442 447 460 468 471 477 509 514
##   1    2    1    1    1    2    1    1    1    1    1    1    1    1    1    1
## 530 587 640
##   1    1    1
##
##   Min. 1st Qu.  Median      Mean 3rd Qu.     Max.
##   0.00    5.00   12.00   19.89   24.00   640.00
##
## includes extended item information - examples:
##               labels
## 1  *Boombox Ipod Classic
## 2 *USB Office Mirror Ball
## 3
##

```

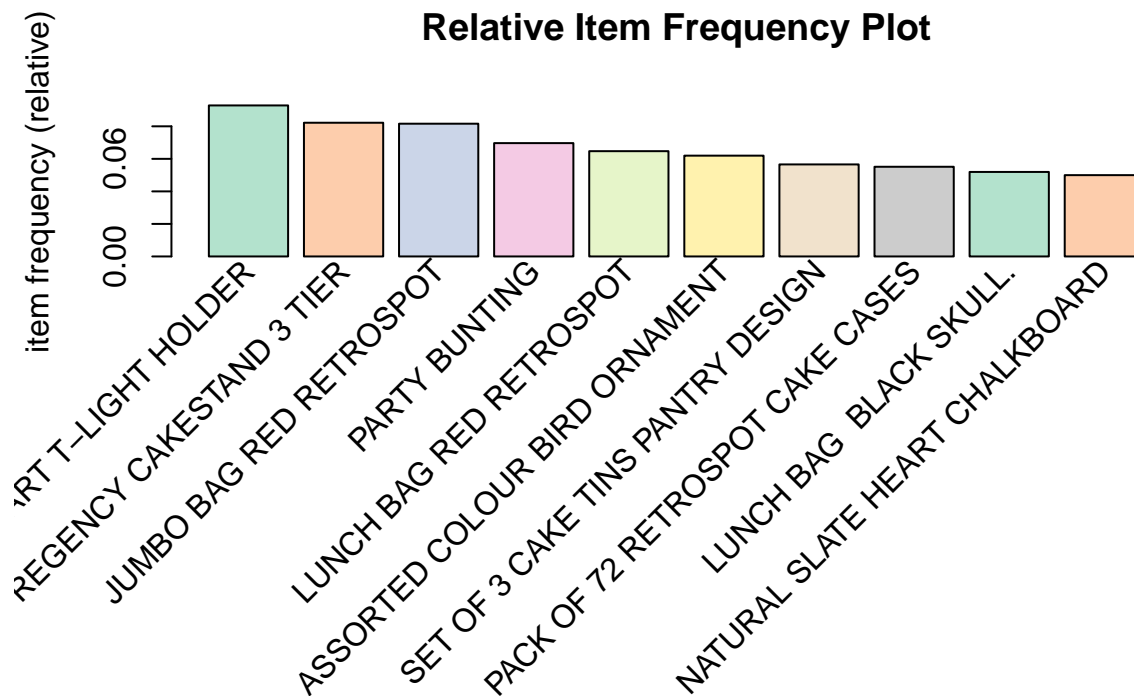
```

# Irei plotar os 10 produtos mais comprados
itemFrequencyPlot(transacao,topN=10,type="absolute",col=brewer.pal(8,'Pastel2'), main="Absolute Item Fr

```



```
itemFrequencyPlot(transacao,topN=10,type="relative",col=brewer.pal(8,'Pastel12'), main="Relative Item Fr
```



## Modelagem

Na 1ª modelagem os resultados foram: 100% das vezes que um cliente comprou o produto 'BILLBOARD FONTS DESIGN' ele também comprou WRAP. Já 93% das vezes que um cliente comprou o wrap ele também comprou o 'BILLBOARD FONTS DESIGN'.

Na 2ª modelagem para o produto WHITE HANGING HEART T-LIGHT HOLDER os resultados foram: ele está muito associado com produtos que parecem ser da mesma "família" em 80% quem com o pink e o red hanging heart t-light holder também comprou o white. 83% quem comprou só o red também comprou o white.

```
associacao = apriori(transacao, parameter = list(supp=0.001, conf=0.8,maxlen=10))
```

```
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
##          0.8    0.1    1 none FALSE                TRUE         5   0.001    1
## maxlen target  ext
##          10   rules TRUE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
##       0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
```

```
## Absolute minimum support count: 20
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[8775 item(s), 20769 transaction(s)] done [0.19s].
## sorting and recoding items ... [2656 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10

## Warning in apriori(transacao, parameter = list(supp = 0.001, conf = 0.8, :
## Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!

## done [3.05s].
## writing ... [3356918 rule(s)] done [1.47s].
## creating S4 object ... done [1.96s].
```

```
# vamos ver as associações
inspect(associacao[1:10])
```

##	lhs	rhs	support	confidence
## [1]	{BILLBOARD FONTS DESIGN}	=> {WRAP}	0.001444460	1.0000000
## [2]	{WRAP}	=> {BILLBOARD FONTS DESIGN}	0.001444460	0.9375000
## [3]	{SET/4 BLUE FLOWER CANDLES IN BOWL}	=> {S/4 PINK FLOWER CANDLES IN BOWL}	0.001059271	0.8148148
## [4]	{BLACK TEA}	=> {SUGAR JARS}	0.002022245	1.0000000
## [5]	{BLACK TEA}	=> {COFFEE}	0.002022245	1.0000000
## [6]	{WOBBLY CHICKEN}	=> {METAL}	0.001492609	1.0000000
## [7]	{WOBBLY CHICKEN}	=> {DECORATION}	0.001492609	1.0000000
## [8]	{SET/20 FRUIT SALAD PAPER NAPKINS}	=> {STRAWBERRY CHARLOTTE BAG}	0.001011122	0.9130435
## [9]	{SET/20 FRUIT SALAD PAPER NAPKINS}	=> {LUNCH BAG CARS BLUE}	0.001011122	0.9130435
## [10]	{SET/20 FRUIT SALAD PAPER NAPKINS}	=> {WOODLAND CHARLOTTE BAG}	0.001011122	0.9130435

```
# Associações para clientes que compraram o
# WHITE HANGING HEART T-LIGHT HOLDER um dos produtos mais comprados.
assoc = apriori(transacao, parameter = list(supp=0.001, conf=0.8), appearance = list(default="lhs", rhs=
```

```
## Apriori
##
## Parameter specification:
## confidence minval smax arem aval originalSupport maxtime support minlen
## 0.8 0.1 1 none FALSE TRUE 5 0.001 1
## maxlen target ext
## 10 rules TRUE
##
## Algorithmic control:
## filter tree heap memopt load sort verbose
## 0.1 TRUE TRUE FALSE TRUE 2 TRUE
##
## Absolute minimum support count: 20
##
## set item appearances ...[1 item(s)] done [0.00s].
## set transactions ...[8775 item(s), 20769 transaction(s)] done [0.16s].
## sorting and recoding items ... [2656 item(s)] done [0.01s].
## creating transaction tree ... done [0.01s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10
```

```
## Warning in apriori(transacao, parameter = list(supp = 0.001, conf = 0.8), :
## Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!
```

```
## done [3.01s].
## writing ... [71 rule(s)] done [0.18s].
## creating S4 object ... done [0.06s].
```

```
inspect(head(assoc))
```

	lhs	rhs	support	confidence
## [1]	{PINK HANGING HEART T-LIGHT HOLDER, RED HANGING HEART T-LIGHT HOLDER}	=> {WHITE HANGING HEART T-LIGHT HOLDER}	0.001155568	0.800000
## [2]	{IVORY WICKER HEART LARGE, RED HANGING HEART T-LIGHT HOLDER}	=> {WHITE HANGING HEART T-LIGHT HOLDER}	0.001203717	0.833333
## [3]	{CREAM CUPID HEARTS COAT HANGER, RED HANGING HEART T-LIGHT HOLDER}	=> {WHITE HANGING HEART T-LIGHT HOLDER}	0.001011122	0.807692
## [4]	{RED WOOLLY HOTTIE WHITE HEART., WHITE METAL LANTERN}	=> {WHITE HANGING HEART T-LIGHT HOLDER}	0.001011122	0.875000
## [5]	{WHITE METAL LANTERN, WOODEN FRAME ANTIQUE WHITE}	=> {WHITE HANGING HEART T-LIGHT HOLDER}	0.001155568	0.827586
## [6]	{FINE WICKER HEART, ZINC METAL HEART DECORATION}	=> {WHITE HANGING HEART T-LIGHT HOLDER}	0.001251866	0.812500