INF0613 – Aprendizado de Máquina Não Supervisionado

Trabalho 1 - Regras de Associação

Nicole Nogueira

Rodolfo Dalla Costa

Neste primeiro trabalho vamos minerar Regras de Associação em uma base de dados que contém as vendas de uma padaria. A base de dados está disponível na página da disciplina no Moodle (arquivo `bakery.csv`).

Atividade 0 – Configurando o ambiente

Antes de começar a implementação do seu trabalho configure o *workspace* e importe todos os pacotes:

```
# Adicione os demais pacotes usados
# Bibliotecas usadas neste trabalho:
library(arules)
library(arulesViz)

# Configurando ambiente de trabalho:
setwd("/Users/rodolfodc/Documents/mineracao-dados-complexos/homeworks/inf-0611-0612/inf0613")
```

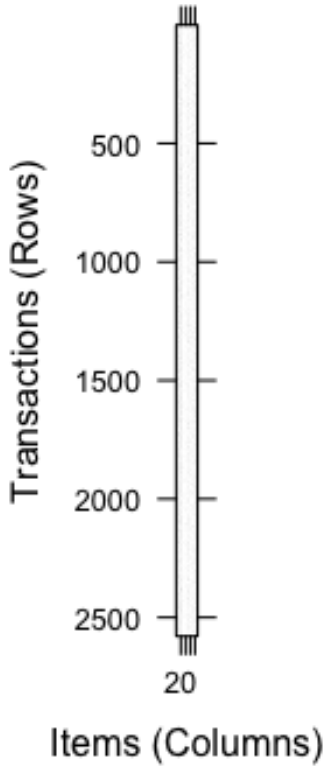
Atividade 1 – Análise Exploratória da Base de Dados (*3,0 pts*)

Dado um caminho para uma base de dados, leia as transações e faça uma análise Exploratória sobre elas. Use as funções `summary`, `inspect` e `itemFrequencyPlot`. Na função `inspect` limite sua análise às 10 primeiras transações e na função `itemFrequencyPlot` gere um gráfico com a frequência relativa dos 30 itens mais frequentes.

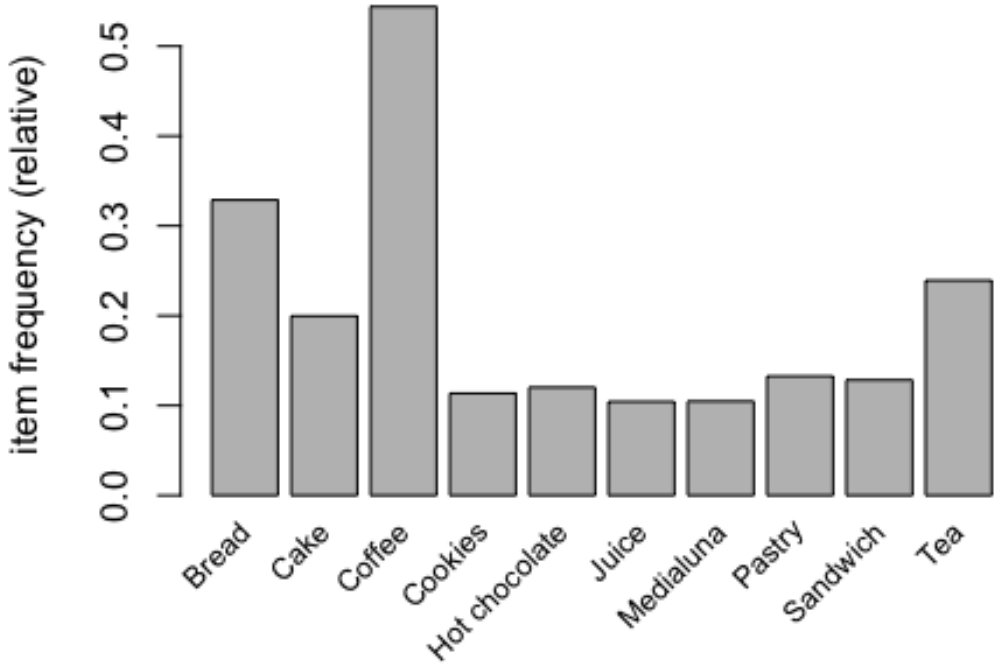
```
# Ler transações
transacoes <- read.transactions("bakery.csv", format="basket", sep=",")

# Visualizando transações
inspect(transacoes[1:10])
```

```
##      items
## [1]  {Coffee,Vegan mincepie}
## [2]  {Farm House,Muffin,Tea}
## [3]  {Bread,Ellas Kitchen Pouches,Jam,Juice,Muffin}
## [4]  {Bread,Juice,Salad,Sandwich}
## [5]  {Cake,Coffee,Sandwich,Smoothies,Soup}
```

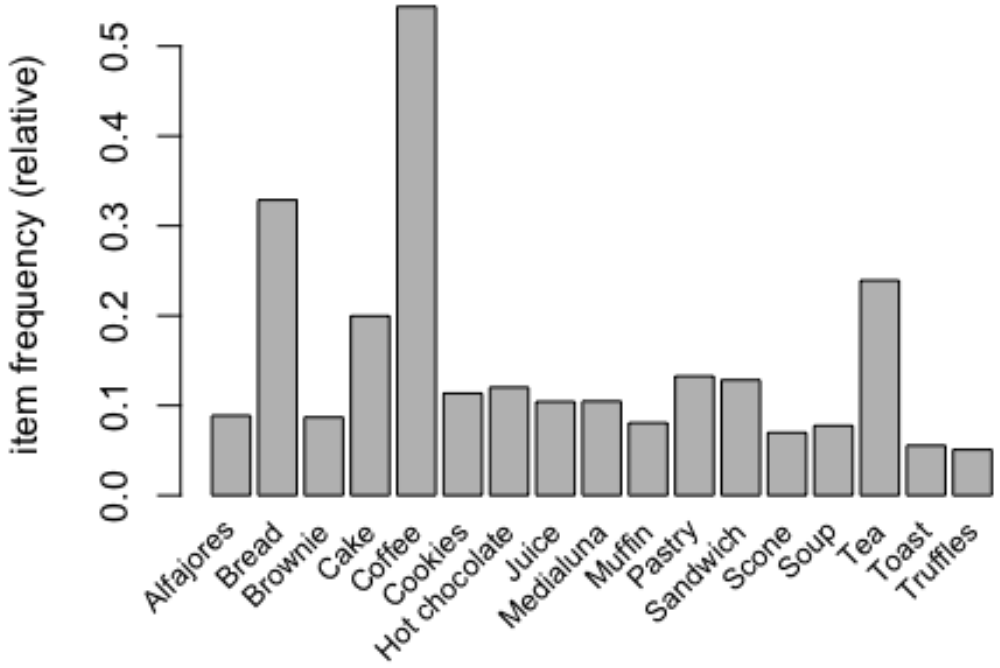
```
## [6]  {Bread,Medialuna}
## [7]  {Chocolates,Coffee,Tea}
## [8]  {Alfajores,Brownie,Medialuna}
## [9]  {Alfajores,Coffee,Fudge}
## [10] {Bread,Pastry}

# Sumário da base
summary(transacoes)

## transactions as itemMatrix in sparse format with
##  2579 rows (elements/itemsets/transactions) and
##  91 columns (items) and a density of 0.0352
##
## most frequent items:
##  Coffee   Bread     Tea    Cake  Pastry (Other)
##    1403     848     617     515     342    4532
##
## element (itemset/transaction) length distribution:
## sizes
##    1    2    3    4    5    6    7    8    9   10
##   20  664 1041  591  189   52   15    4    2    1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0     2.0     3.0     3.2     4.0    10.0
##
## includes extended item information - examples:
##                     labels
## 1 Afternoon with the baker
## 2               Alfajores
## 3          Argentina Night

# Analisando a frequência dos itens
image(transacoes)
```




```
itemFrequencyPlot(transacoes, support = 0.1, cex.names = 0.8)
```




```
itemFrequencyPlot(transacoes, support=0.05, cex.names=0.8)
```



Análise

a) Descreva a base de dados discutindo os resultados das funções acima.

Resposta:

A base de dados eh composta por um conjunto de produtos consumiveis, como cafe, pao, chocolate quente ou cha, ao todo sao 91 itens diferentes. Nenhum registro contem itens repetidos, ao todo, sao 2579 transacoes (cada linha eh uma transacao), o mais comum eh ter um conjunto de 3 itens presente nas transacoes (esse padrao ocorre 1041 vezes), e eh possivel encontar um conjunto de ate 10 itens, porem, este ultimo, numa unica transacao. Os 3 produtos mais frequentes sao Coffee, Bread e Tea, sendo que suas frequencias sao 1403, 808 e 617 respectivamente. O conjunto tem uma densidade de 0.0352, ou seja, 3,52% de celulas nao-zero na matriz. E alem disso o numero medio de transacoes possui 3.02 itens.

b) Ao gerarmos o gráfico de frequências, temos uma representação visual de uma informação já presente no resultado da função `summary`. Contudo, esse gráfico nos dá uma visão mais ampla da base. Assim podemos ver a frequência de outros itens em relação aos 10 mais frequentes. Quais informações podemos obter a partir desse gráfico (e da análise anterior) para nos ajudar na extração de regras de associação

com o algoritmo `apriori`? Isto é, como a frequência dos itens pode afetar os parâmetros de configuração do algoritmo `apriori`?

Resposta:

Os graficos de frequencia revelam o vies presente na base de dados. Neste caso, o grafico indica por exemplo, que o item cafe foi muito mais comprado que qualquer outro item, ou seja, o algoritmo `apriori` tendenciosamente encontrara muitas regras que envolvem conjuntos com item cafe e que terao muitos resultados diferentes, portanto o algoritmo precisara ser ajustado para ter, por exemplo, um baixo suporte minimo, afim de evitar os problemas relacionados ao vies. Alem disso, ja se espera que qualquer regras que correlacionem com cafe, merecem uma atencao redobrada ja que o cafe eh um item extremamente frequente e, portanto, sua confianca sera alta.

Atividade 2 – Minerando Regras (*3,5 pts*)

Use o algoritmo `apriori` para minerar regras na base de dados fornecida. Experimente com pelo menos *3 conjuntos* de valores diferentes de suporte e confiança para encontrar regras de associação. Imprima as cinco regras com o maior suporte de cada conjunto escolhido. Lembre-se de usar seu conhecimento sobre a base, obtido na questão anterior, para a escolha dos valores de suporte e confiança.

```
# Conjunto 1: suporte = 0.1   e confiança = 0.5
regras1 <- apriori(transacoes, parameter=list(supp=0.1, conf=0.5))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.5    0.1    1 none FALSE            TRUE       5     0.1      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: 257
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 done [0.00s].
## writing ... [2 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
# Conjunto 2: suporte = 0.05   e confiança = 0.08
regras2 <- apriori(transacoes, parameter=list(supp=0.05, conf=0.08))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.08    0.1    1 none FALSE            TRUE       5    0.05      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: 128
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
## sorting and recoding items ... [17 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 done [0.00s].
## writing ... [37 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
# Conjunto 3: suporte = 0.0006   e confiança = 0.7
regras3 <- apriori(transacoes, parameter=list(supp=0.0006, conf=0.7))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5   6e-04      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: 1
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[91 item(s), 2579 transaction(s)] done [0.00s].
## sorting and recoding items ... [83 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [524 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```r
inspect(sort(regras1, by=c("support"), decreasing = TRUE))
```

```
##     lhs       rhs      support confidence coverage lift count
## [1] {}     => {Coffee} 0.544   0.544      1.0      1.00 1403
## [2] {Cake} => {Coffee} 0.112   0.559      0.2      1.03  288
```

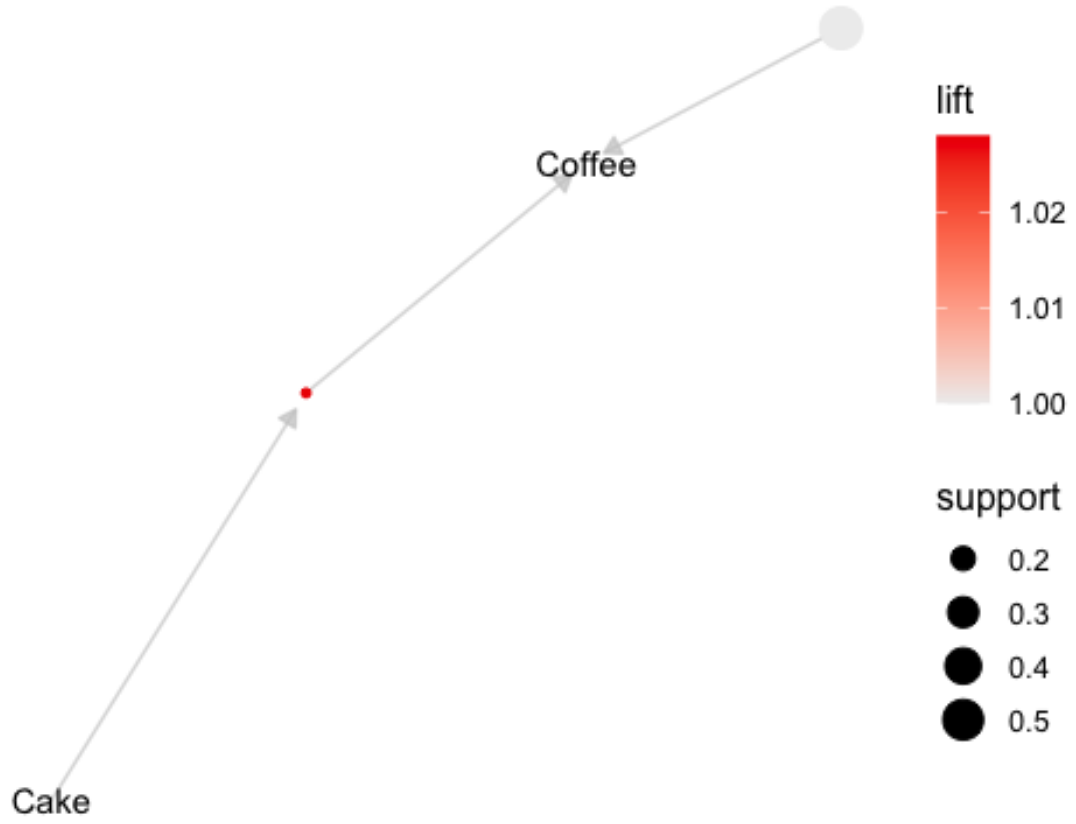
```
inspect(sort(regras2, by=c("support"), decreasing = TRUE)[1:5])
```

```
##     lhs        rhs      support confidence coverage lift  count
## [1] {}      => {Coffee} 0.544   0.544      1.000    1.000 1403
## [2] {}      => {Bread}  0.329   0.329      1.000    1.000  848
## [3] {}      => {Tea}    0.239   0.239      1.000    1.000  617
## [4] {}      => {Cake}   0.200   0.200      1.000    1.000  515
## [5] {Bread} => {Coffee} 0.154   0.468      0.329    0.861  397
```

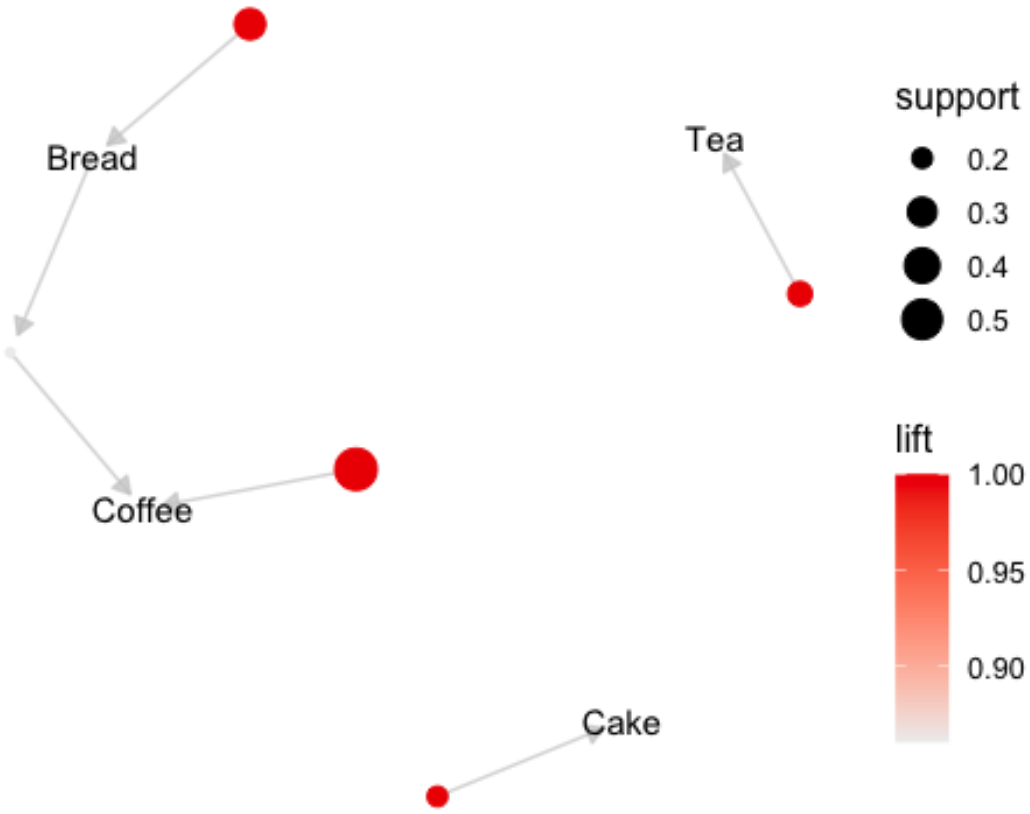
```
inspect(sort(regras3, by=c("support"), decreasing = TRUE)[1:5])
```

```
##     lhs                              rhs      support confidence coverage
lift
## [1] {Toast}                       => {Coffee} 0.03994 0.720      0.05545
1.32
## [2] {Keeping It Local}            => {Coffee} 0.00969 0.781      0.01241
1.44
## [3] {Extra Salami or Feta}        => {Coffee} 0.00698 0.900      0.00775
1.65
## [4] {Extra Salami or Feta}        => {Salad}  0.00620 0.800      0.00775
31.26
## [5] {Extra Salami or Feta,Salad} => {Coffee} 0.00543 0.875      0.00620
1.61
##     count
## [1] 103
## [2]  25
## [3]  18
## [4]  16
## [5]  14
```

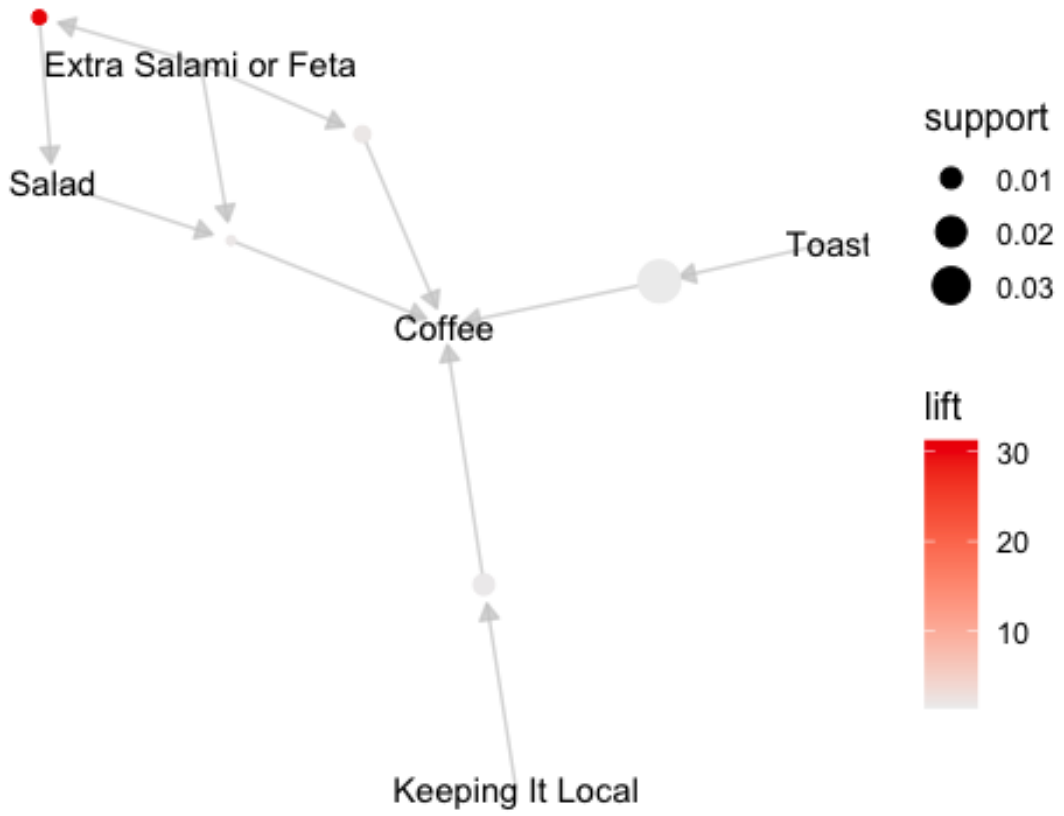
```
plot(sort(regras1, by=c("support"), decreasing = TRUE), method="graph")
```




```
plot(sort(regras2, by=c("support"), decreasing = TRUE)[1:5], method="graph")
```




```
plot(sort(regras3, by=c("support"), decreasing = TRUE)[1:5], method="graph")
```



Análises

a) Quais as regras mais interessantes geradas a partir dessa base? Justifique.

Resposta: 1. {Toast} => {Coffee} 2. {Keeping It Local} => {Coffee}
3. {Extra Salami or Feta} => {Coffee}
4. {Extra Salami or Feta} => {Salad}
5. {Extra Salami or Feta,Salad} => {Coffee}

Acima estao listadas as 5 primeiras regras do conjunto 3. O conjunto 3 foi o que apresentou uma gama maior e mais interessante de regras (ao todo 524 regras). Apesar de manter muitas regras de associacao com o item coffee, eh possivel observar na regra 4 por exemplo uma perspectiva distinta do vies do dataset pois nao envolve nenhum dos 3 itens mais frequentes e portanto indica que esta regra eh de fato um padrao que pode ser minerado da base. Como ja era esperado, a mineracao utilizando o `apriori` teve um desempenho mais interessante quando o suporte minimo considerado foi alterado para um valor muito baixo, afim de evitar perder itens raros ou evidenciar regras obvias ou enviesadas, mas a confianca foi mantida num patamar mais alto (70%) o que permitiu a mineracao de mais regras e garantem que essas regras estejam presentes em mais conjuntos.

Atividade 3 – Medidas de Interesse (*3,5 pts*)

Vimos na aula que, mesmo após as podas do algoritmo `apriori`, ainda temos algumas regras com características indesejáveis como redundâncias e dependência estatística negativa. Também vimos algumas medidas que nos ajudam a analisar melhor essas regras como o lift, a convicção e a razão de chances. Nesta questão, escolha um dos conjuntos de regras geradas na atividade anterior e o analise usando essas medidas. Compute as três medidas para o conjunto escolhido com a função `interestMeasure` e experimente ordenar as regras com cada uma das novas medidas.

```
# Compute as medidas de interesse
medidas_interesse <- interestMeasure(regras3, c("conviction", "lift",
"oddsRatio"), transacoes)

# Apresente as regras ordenadas por lift
sorted_lift <- sort(regras3, by=c("lift"), decreasing=TRUE)

# Apresente as regras ordenadas por convicção
sorted_conviction <- sort(medidas_interesse$conviction, decreasing = TRUE,
index.return = TRUE, na.last=TRUE)$ix

# Apresente as regras ordenadas por razão de chances
sorted_oddsRation <- sort(medidas_interesse$oddsRatio, decreasing = TRUE,
index.return = TRUE, na.last=TRUE)$ix
head(sort(medidas_interesse$oddsRatio, decreasing=TRUE))
```

```
## [1] 1542.0  200.7  168.9   87.9   87.9   57.4
```

```
inspect(sorted_lift[1:10])
```

```
##      lhs                        rhs                      support confidence
coverage  lift count
## [1]  {Postcard}              => {Tshirt}               0.001163       0.75
0.001551 241.8     3
## [2]  {Bread,
##       Salad,
##       Scandinavian}          => {Extra Salami or Feta} 0.000775       1.00
0.000775 128.9     2
## [3]  {Juice,
##       Salad,
##       Spanish Brunch}        => {Extra Salami or Feta} 0.000775       1.00
0.000775 128.9     2
## [4]  {Bread,
##       Cake,
##       Salad}                 => {Extra Salami or Feta} 0.000775       1.00
0.000775 128.9     2
## [5]  {Coffee,
##       Juice,
##       Salad,
##       Spanish Brunch}        => {Extra Salami or Feta} 0.000775       1.00
```

```
0.000775 128.9     2
## [6]  {Bread,
##       Cake,
##       Coffee,
##       Salad}               => {Extra Salami or Feta} 0.000775       1.00
0.000775 128.9     2
## [7]  {Hack the stack}       => {Art Tray}             0.000775       1.00
0.000775  95.5     2
## [8]  {Extra Salami or Feta,
##       Scandinavian}         => {Salad}                0.000775       1.00
0.000775  39.1     2
## [9]  {Extra Salami or Feta,
##       Juice}                => {Salad}                0.001163       1.00
0.001163  39.1     3
## [10] {Extra Salami or Feta,
##       Sandwich}             => {Salad}                0.001163       1.00
0.001163  39.1     3

inspect(regras3[sorted_conviction][1:10])

##      lhs                              rhs             support confidence
## [1]  {Alfajores,Spanish Brunch}    => {Tea}           0.00233 0.857
## [2]  {Extra Salami or Feta}        => {Salad}         0.00620 0.800
## [3]  {Extra Salami or Feta}        => {Coffee}        0.00698 0.900
## [4]  {Alfajores,Cookies,Tea}       => {Juice}         0.00155 0.800
## [5]  {Coffee,Extra Salami or Feta} => {Salad}         0.00543 0.778
## [6]  {Nomad bag}                   => {Bread}         0.00194 0.833
## [7]  {Postcard}                    => {Tshirt}        0.00116 0.750
## [8]  {Juice,Pick and Mix Bowls}    => {Mineral water} 0.00116 0.750
## [9]  {Bread,Scone,Truffles}        => {Mineral water} 0.00116 0.750
## [10] {Bread,Mineral water,Scone}   => {Truffles}      0.00116 0.750
##      coverage lift   count
## [1]  0.00271    3.58  6
## [2]  0.00775   31.26 16
## [3]  0.00775    1.65 18
## [4]  0.00194    7.67  4
## [5]  0.00698   30.39 14
## [6]  0.00233    2.53  5
## [7]  0.00155  241.78  3
## [8]  0.00155   21.98  3
## [9]  0.00155   21.98  3
## [10] 0.00155   14.77  3

inspect(regras3[sorted_oddsRation][1:10])

##      lhs                              rhs             support confidence
## [1]  {Postcard}                    => {Tshirt}        0.00116 0.750
## [2]  {Extra Salami or Feta}        => {Salad}         0.00620 0.800
## [3]  {Coffee,Extra Salami or Feta} => {Salad}         0.00543 0.778
## [4]  {Juice,Pick and Mix Bowls}    => {Mineral water} 0.00116 0.750
## [5]  {Bread,Scone,Truffles}        => {Mineral water} 0.00116 0.750
```

```
## [6]  {Bread,Mineral water,Scone}    => {Truffles}      0.00116 0.750
## [7]  {Alfajores,Cookies,Tea}        => {Juice}         0.00155 0.800
## [8]  {Coffee,Cookies,Juice,Tea}     => {Alfajores}     0.00116 0.750
## [9]  {Chocolates,Hot chocolate}     => {Juice}         0.00116 0.750
## [10] {Alfajores,Coffee,Cookies,Tea} => {Juice}         0.00116 0.750
##      coverage lift   count
## [1]  0.00155  241.78  3
## [2]  0.00775   31.26 16
## [3]  0.00698   30.39 14
## [4]  0.00155   21.98  3
## [5]  0.00155   21.98  3
## [6]  0.00155   14.77  3
## [7]  0.00194    7.67  4
## [8]  0.00155    8.45  3
## [9]  0.00155    7.19  3
## [10] 0.00155    7.19  3
```

```
# Verificando se existe alguma medida de conviccao dentro do consideado ideal (valor >1 e <=5)
any(medidas_interesse[medidas_interesse$conviction >= 1 &&
medidas_interesse$conviction <= 5,]$conviction,na.rm = TRUE)
```

```
## [1] FALSE
```

Análise

a) Quais as regras mais interessantes do conjunto? Justifique.

Resposta: Para os conjuntos, o que destaca as regras mais interessantes, eh a metrica do Lyft, que trata a independencia de ambos conjuntos da regra. ## lyft

1. {Postcard} => {Tshirt} 0.001163 0.750 0.001551 241.78 3
2. {Bread,Salad,Scandinavian} => {Extra Salami or Feta}
3. {Juice,Salad,Spanish Brunch} => {Extra Salami or Feta}
4. {Bread,Cake,Salad} => {Extra Salami or Feta}
5. {Coffee,Juice,Salad,Spanish Brunch} => {Extra Salami or Feta}
6. {Bread,Cake,Coffee,Salad} => {Extra Salami or Feta}
7. {Hack the stack} => {Art Tray}
8. {Bread,Extra Salami or Feta} => {Salad}

Acima estao listadas as 8 primeiras regras decorrentes da ultima inspecao nas regras do conjunto 3. O conjunto 3 foi eleito como o melhor suporte/confianca pois elabora 524 regras e traz perspectivas interessantes de conjuntos de itens, o que foi notado com menor frequencia nos outro conjuntos As 8 rrevelam uma perspectiva de confianca na regra. Elas sao bem interessantes pois, por exemplo, percebe-se na primeira regra, o perfil de compras de um turista, a compra de um cartao postal leva a compra de uma camisa. As regras 2 a 6 sao interessantes pois trazem regras de conjuntos que nao possuem os dois itens mais frequentes (bread e coffee), como e o caso da regra 3. Desse modo, as 8 regras selecionadas acima

sao muito interessantes pois revelam perfis de consumo, como o caso do turista, ou revelam tendencias de consumidores comuns. Alem disso, a ordenacao de regras por conviccao nao trouxe resultados interessantes dentro dos patamares esperados e o conjunto de regras ordenado por razao de chances retornou uma perspectiva parecida, um ponto interessante do ordenacao por razao de chances, eh que essa ordenacao trouxe em primeiro mais regras que envolvem menos itens com alta frequencia no dataset. Desse modo, a ordenacao por Lyft demonstra trazer um perspectiva de perfis de consumo mais interessante diante das outras medidas de interesse, ja que elenca regras que fogem ao vies logo nas primeiras 8 regras e revelam composicoes de conjuntos com menor presencao de itens muito frequentes.