



Faculdade de Economia, Administração e Contabilidade

Machine Learning

Laboratório de Análise de Dados

Análise de Regras de Associação

com R e RStudio

Prof. Antonio Geraldo da Rocha Vidal

2023

Sumário

Introdução	4
Regras de Associação	4
Conceitos Básicos de Análise de Regras de Associação	5
Objetivo da Análise de Regras de Associação	7
Algoritmo APRIORI	7
Implementando Regras de Associação usando R	10
O Conjunto de Dados	10
Descrição do Dataset.....	11
Carregando Bibliotecas do R	11
Pré-processamento de dados	12
Gerando Regras.....	17
Limitar o número e o tamanho das regras.....	18
Removendo Regras Redundantes	18
Encontrando Regras Relacionadas a Itens Fornecidos.....	18
Visualizando Regras de Associação	20
Gráfico de Dispersão	20
Gráfico de Dispersão Interativo	21
Visualizações Baseadas em Gráficos	22
Representação de Regra Individual.....	22
Etapa Final: Relatório de Elaboração do Laboratório	23
Conclusão	23
Referências.....	24

Você deve entregar um relatório com os resultados das etapas elaboradas neste laboratório no **e-Disciplinas**, para formatá-lo siga estas orientações:

1. Crie um documento Word e identifique-o com o nome do laboratório, data de elaboração e o seu nome ou do grupo que o elaborou;
2. Crie um tópico para cada resultado que você considerar relevante (manipulação de dados ou resultado de algum processamento) identificando-o com um título e uma breve explicação. Os resultados podem ser imagens de gráficos gerados ou de listas de valores ou dados de resultados obtidos. Não devem ser incluídos os *scripts* ou instruções de processamento utilizados, inclua apenas os resultados que você considerar relevantes.
3. No final do relatório crie um último tópico denominado “Conclusões” e elabore comentários, sugestões e conclusões sobre o que você pode aprender com a elaboração deste laboratório.

Esta apostila introdutória pode conter erros, falhas ou imprecisões. Se você identificar algum problema por favor informe através do e-mail vidal@usp.br para que a correção possa ser providenciada.

Esta apostila não é autoral, tem objetivo estritamente didático como material de apoio para a disciplina. Foi desenvolvida através da compilação dos diversos textos e materiais citados na bibliografia.



Obrigado!

Introdução

Você é um cientista de dados (ou está se tornando um!), e recebe um empreendedor que possui uma loja de varejo. Seu cliente fornece dados de todas as transações que consistem em itens comprados na loja por vários clientes ao longo de um período e solicita que você use esses dados para ajudar a impulsionar seus negócios. Neste laboratório você estudará como poderá ajudar os varejistas a impulsionar os negócios prevendo os itens que os clientes compram juntos.

Seu cliente usará suas descobertas não apenas para alterar/atualizar/adicionar itens no estoque, mas também irá usá-los para alterar o layout da loja física ou, em vez disso, de uma loja online. Para encontrar resultados que ajudarão o seu cliente, você usará **Análise Associação**, uma técnica que utiliza **regras de associação** para analisar os dados das transações do negócio.

Neste laboratório você vai aprender:

1. O que é uma regra de associação
2. O que é o algoritmo APRIORI?
3. Como implementar Análise de Associação usando a linguagem R com visualizações.

Regras de Associação

A técnica de Regra de Associação deve ser usada quando você deseja encontrar uma associação entre diferentes objetos em um conjunto, encontrar **padrões** frequentes em um banco de dados de transações, ou qualquer outro repositório de dados. As aplicações da análise de regras de associação são encontradas em marketing, análise de cesta de vendas (ou análise de cesta de mercado) no varejo, agrupamento e classificação. Ela pode dizer-lhe que itens os clientes compram frequentemente juntos, gerando um conjunto de regras chamadas **Regras de Associação**.

Fornece uma saída com regras no formato: **se ocorre isso, então** Pode-se utilizar as regras de associação para numerosas estratégias de marketing:

- Alterar o layout da loja de acordo com as tendências de itens associados
- Analisar o comportamento do Cliente
- Projetar catálogos de itens
- Marketing cruzado em lojas online (quem compra esse item também compra esses outros...)
- Quais são os itens que os clientes tendem a comprar em conjunto
- Enviar e-mails personalizados para os clientes para aumentar as vendas
- Etc.

Considere o seguinte exemplo:

ID	Items
1	{Bread, Milk}
2	{Bread, Diapers, Beer, Eggs}
3	{Milk, Diapers, Beer, Cola}
4	{Bread, Milk, Diapers, Beer}
5	{Bread, Milk, Diapers, Cola}
...	...

market
basket
transactions

{Diapers, Beer} Example of a frequent itemset

{Diapers} → {Beer} Example of an association rule

Dado um conjunto de transações de vendas, você pode analisar as transações numeradas de 1 a 5. Cada transação mostra os itens comprados nessa transação. Você pode ver que fraldas (*Diapers*) são compradas com cerveja (*Beer*) em três transações. Da mesma forma pão (*Bread*) é comprado com leite (*Milk*) em três transações tornando-os ambos os conjuntos de itens mais frequentes. As regras de associação são dadas no formato abaixo:

$$A \Rightarrow B[\text{Suporte}, \text{Confiança}]$$

A parte anterior $A \Rightarrow$ é referida como **SE (antecedente)** e a parte posterior **B[Suporte, Confiança]** é referida como **Então (consequente)**. Onde **A** e **B** são conjuntos de itens contidos nos dados de transações. **A** e **B** são conjuntos de disjunção. Por exemplo:

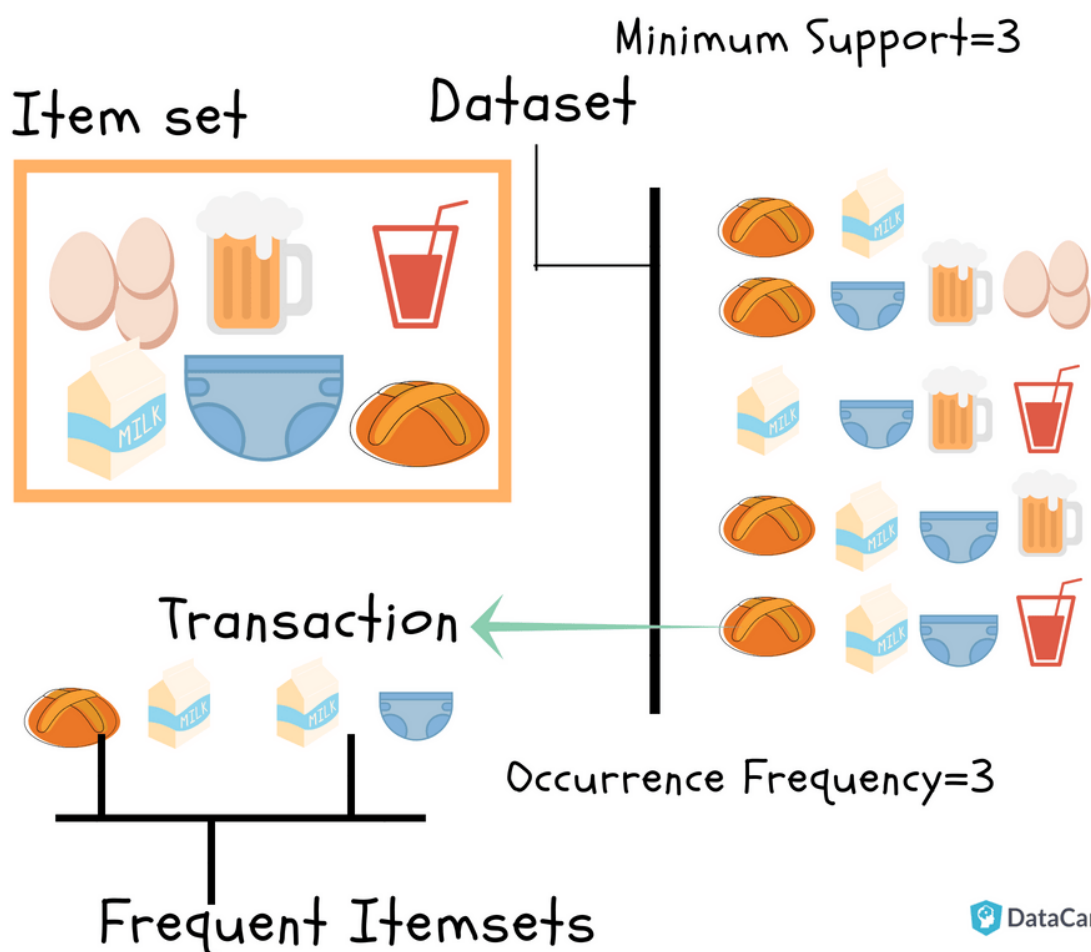
$$\text{Computador} \Rightarrow \text{Software Antivírus} [\text{Apoio} = 20\%, \text{Confiança} = 60\%]$$

A regra acima diz:

1. 20% das transações mostram que software antivírus é comprado em conjunto com a compra de um computador (este é o Suporte);
2. 60% dos clientes que compram software antivírus, o compraram com a compra de um computador (esta é a Confiança).

Na seção a seguir, você aprenderá sobre os conceitos básicos da análise de regras de associação.

Conceitos Básicos de Análise de Regras de Associação



1. **Itens:** uma coleção de um ou mais itens. K-itemset significa um conjunto de k itens.
2. **Contagem de suporte:** frequência de ocorrência de um conjunto de itens.
3. **Suporte (s):** fração de transações que contêm o conjunto de itens.

$$Support(X) = \frac{frequency(X)}{N}$$

Para a Regra $A \Rightarrow B$, o Suporte é dado por:

$$Support(A \Rightarrow B) = \frac{frequency(A, B)}{N}$$

$$Support(A \Rightarrow B) = P(A \cup B) = \frac{n(A \cup B)}{N}$$

Nota: $P(A \cup B)$ é a probabilidade de A e B ocorrendo em conjunto. P denota probabilidade.

Em palavras, é o número de transações com **A** e **B** dividido pelo número total de Transações. **N** é o número total de transações.

Como exercício, tente encontrar o suporte para $Milk \Rightarrow Diaper$.

1. **Confiança (c):** Para uma regra $A \Rightarrow B$ confiança mostra a porcentagem na qual B é comprado com A.

$$Confidence(A \Rightarrow B) = \frac{P(A \cap B)}{P(A)} = \frac{frequency(A, B)}{frequency(A)}$$

$$Confidence(A \Rightarrow B) = \frac{P(A \cup B)}{P(A)} = \frac{n(A \cup B)}{n(A)}$$

O número de transações com A e B dividido pelo número total de transações com A.

$$Confidence(Bread \Rightarrow Milk) = \frac{3}{4} = 0.75 = 75\%$$

Agora encontre a confiança para o $Milk \Rightarrow Diaper$.

$$Confidence(Milk \Rightarrow Diaper) = \frac{3}{4} = 0.75 = 75\%$$

Nota: suporte e confiança medem o quão interessante é a regra de associação. Uma regra é definida pelo mínimo de suporte e limites mínimos de confiança. Esses limiares definidos pelo cliente ajudam a comparar a força da regra, de acordo com sua própria vontade ao a vontade do cliente. Quanto mais próximo do limite, mais a regra é de útil para o cliente.

1. **Conjuntos de itens frequentes:** itemsets cujo suporte é maior ou igual ao limite mínimo de suporte (Min_sup). No exemplo acima Min_sup= 3. Isso é definido pelo usuário ou analista.
2. **Regras fortes:** se uma regra $A \Rightarrow B$ [Apoio, confiança] satisfaz Min_sup e Min_Confidence então é uma regra forte.
3. **Lift:** dá a correlação entre A e B na regra $A \Rightarrow B$. A correlação mostra como um itemset A afeta o itemset B.

$$Lift(A \Rightarrow B) = \frac{Support}{Supp(A)Supp(B)}$$

A e B são independentes se:

$$P(A \cup B) = P(A)P(B)$$

caso contrário dependentes. A associação é dada por:

$$Lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}$$

Assim, quanto maior a associação, maior a chance de A e B ocorrendo juntos.

$$Lift(A \Rightarrow B) = \frac{Support}{Supp(A)Supp(B)}$$

Por exemplo, na regra {Bread} \Rightarrow {Milk}, Lift é calculado como:

$$support(Bread) = \frac{4}{5} = 0.8$$

$$support(Milk) = \frac{4}{5} = 0.8$$

$$Lift(Bread \Rightarrow Milk) = \frac{0.60}{0.80 * 0.80} = 0,94$$

- Se a regra tiver um Lift de 1, então A e B são independentes e nenhuma regra pode ser derivada deles.
- Se o Lift for > 1, a e B são dependentes um do outro, e o grau de que é dado pelo valor de LIFT.
- Se o Lift for < 1, então a presença de A terá efeito negativo sobre B.

Objetivo da Análise de Regras de Associação

Quando você aplica a análise de regras de associação em um determinado conjunto de transações **T** seu objetivo será encontrar todas as regras interessantes, isto é, aquelas com:

1. Suporte maior ou igual ao Suporte Mínimo (*min_support*)
2. Confiância maior ou igual à Confiância Mínima (*min_confidence*)

Algoritmo APRIORI

Nesta parte do laboratório, você entenderá o algoritmo APRIORI, que é executado por trás das bibliotecas do R para análise de regras de associação. Isso irá ajudá-lo a entender mais seus dados e realizar análises com mais clareza.

A análise de regras de associação é vista como uma abordagem em duas etapas:

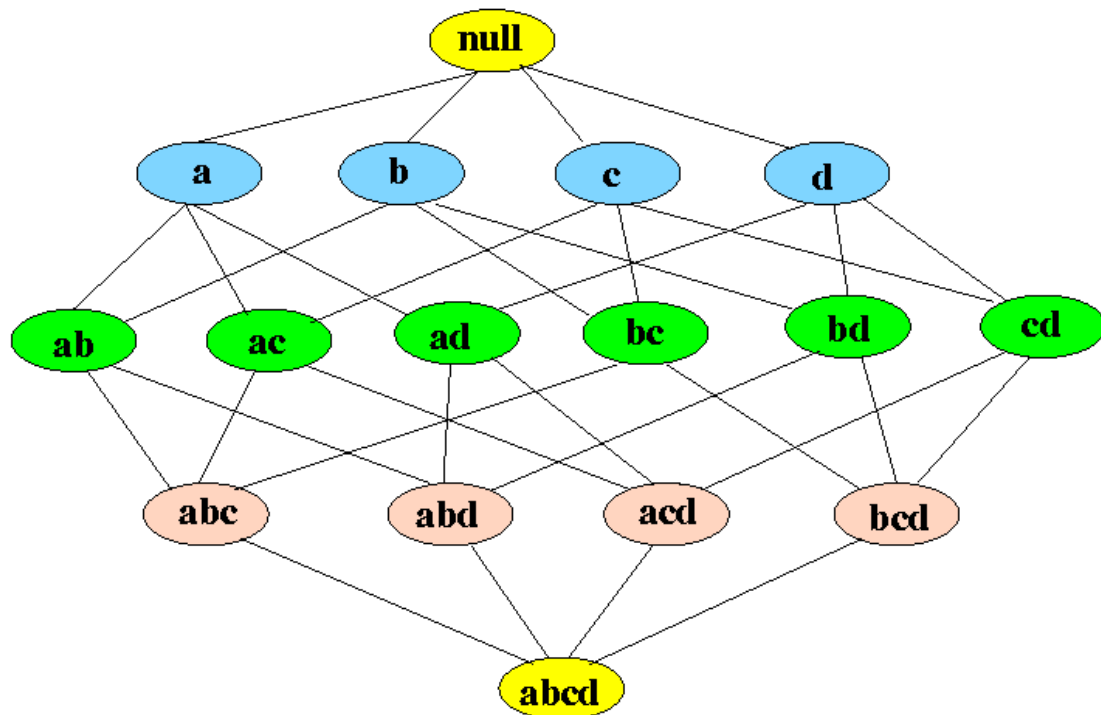
1. **Geração de Itemset frequente:** encontrar todos os conjuntos de itens frequentes com suporte \geq min_support pré-determinado;
2. **Geração de regras:** listar todas as regras de associação de conjuntos de itens frequentes. Calcule o suporte e a confiança para todas as regras. Remover regras que falham nos limiares de min_support e min_confidence.

A geração de Itemset frequente é a etapa computacionalmente mais dispendiosa porque requer uma verificação completa do banco de dados das transações. Entre as etapas acima, a geração de conjunto de itens frequentes é a mais onerosa em termos de uso de recursos computacionais.

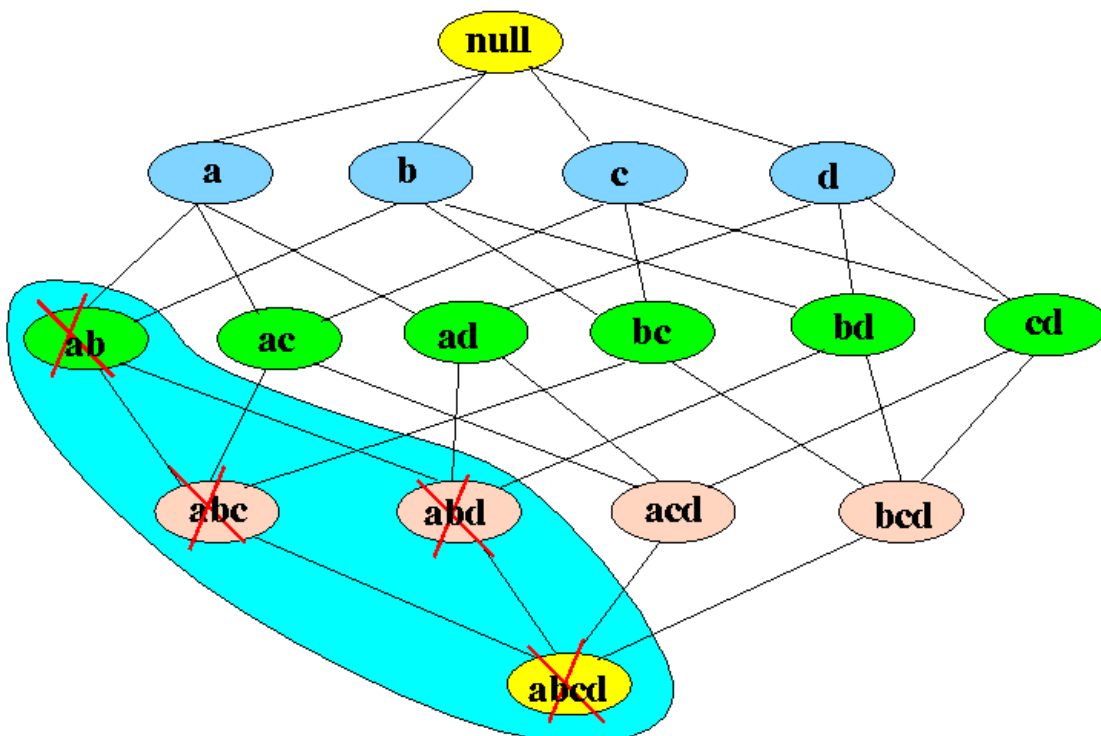
No exemplo conceitual anterior, utilizamos um itemset com apenas 5 transações. Entretanto, em dados de transações do mundo real para o varejo ou outro tipo de aplicação, a quantidade de transações pode exceder GBs e TBs de dados, ou seja, milhões de dados de transações. Para um volume desse tamanho um algoritmo otimizado é necessário para podar os conjuntos de itens pouco frequentes que não ajudarão em etapas posteriores. Para isso é usado o algoritmo APRIORI, que afirma:

“Qualquer subconjunto de um conjunto de itens frequente também deve ser frequente. Ou em outras palavras, nenhum superconjunto de um conjunto de itens não frequente deve ser gerado ou testado.”

Ele pode ser apresentado em uma **estrutura do conjunto de itens** que é uma representação gráfica do princípio do algoritmo APRIORI. Consiste no nó do k-itemset e na relação dos subconjuntos daquele k-itemset.

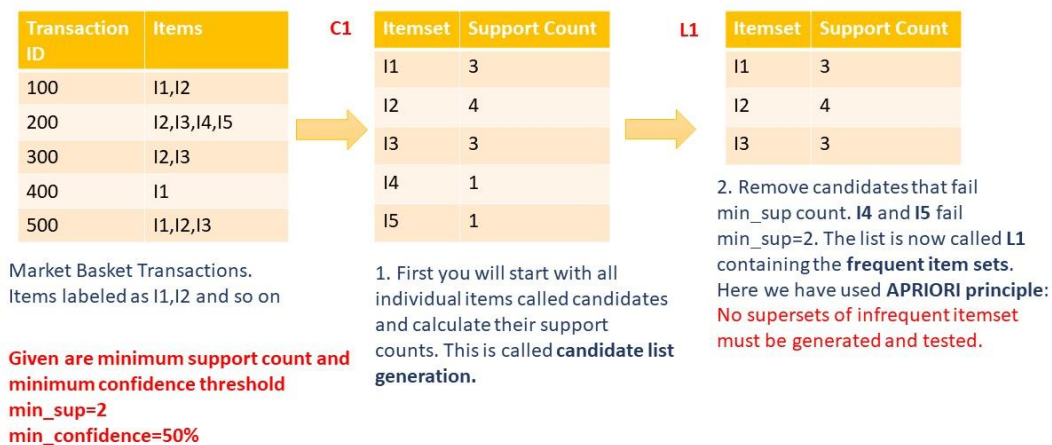


Você pode observar na figura acima que na parte inferior estão todos os itens de dados de transação e, em seguida, você começa a se mover para cima criando subconjuntos até o conjunto nulo. Para d número de itens de tamanho da estrutura se tornará 2^d . Isso mostra como será difícil gerar o itemset frequente encontrando o suporte para cada combinação. A figura a seguir mostra quanto APRIORI ajuda a reduzir o número de conjuntos a serem gerados:

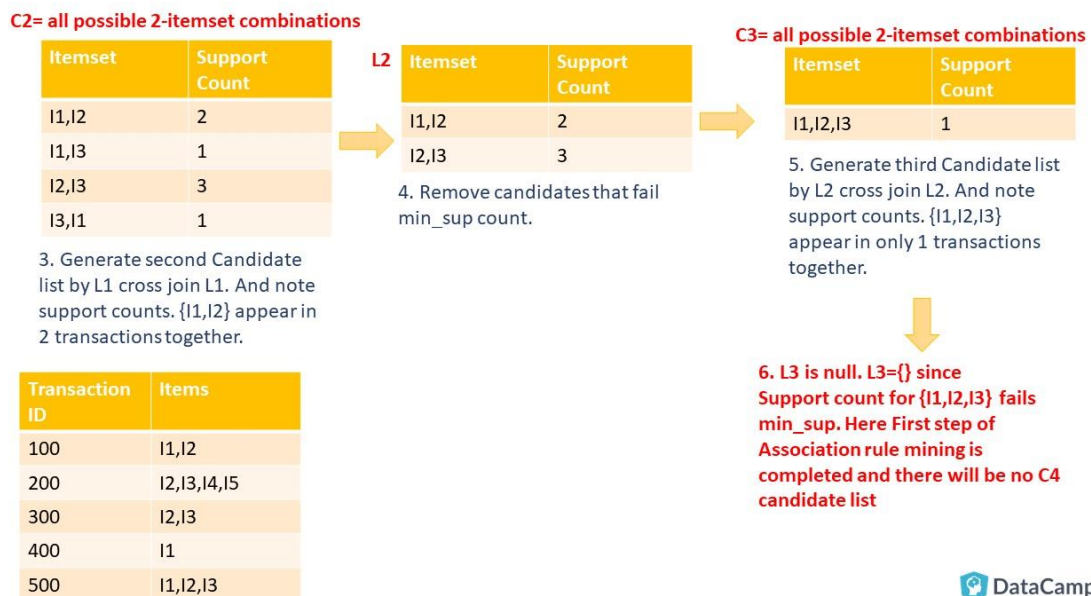


Se o conjunto de itens $\{a, b\}$ é pouco frequente, então não precisamos levar em conta todos os seus supersets. Vamos entender isso com um exemplo.

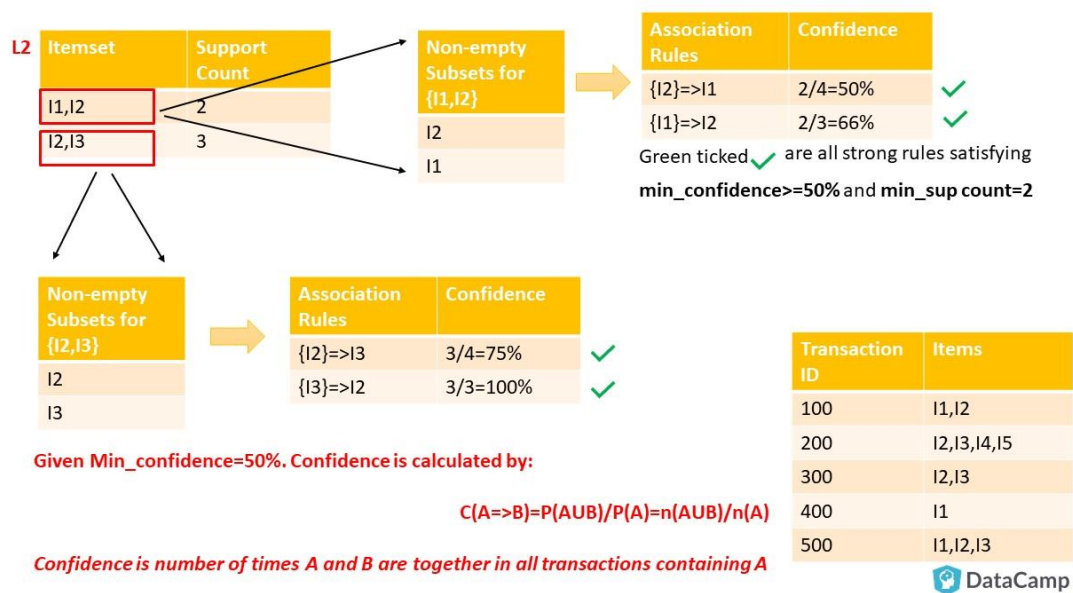
No exemplo a seguir, você verá por que APRIORI é um algoritmo eficaz e gerará regras de associação fortes passo a passo.



Como você pode ver, você começa criando a *Lista de candidatos* para o 1-Itemset que incluirá todos os itens, que estão presentes nos dados da transação, individualmente. Considerando os dados de transações de varejo do mundo real, você pode avaliar quão custosa pode ser essa geração de candidatos. Aqui o algoritmo APRIORI desempenha seu papel e ajuda a reduzir o número da lista de candidatos, e apenas as regras úteis são geradas no final. Nas etapas a seguir, você observará como chegamos ao final da geração do conjunto de itens frequentes, que é a primeira etapa da análise de regra de associação.



Seu próximo passo será listar todos os conjuntos de itens frequentes. Você tomará o último conjunto de itens frequentes não vazio, que neste exemplo é **L2={I1, I2}, {I2, I3}**. Em seguida, você pode analisar todos os subconjuntos não vazios dos conjuntos de itens presentes nessa lista de item frequentes. Prossiga conforme mostra a ilustração a seguir:



Você pode ver acima que há quatro regras fortes. Por exemplo, considere $I2 \Rightarrow I3$ ter confiança igual a 75%, isso significa que 75% das pessoas que compraram $I2$ também compram $I3$.

Agora que você aprendeu como o algoritmo APRIORI funciona, um dos algoritmos mais usados na mineração de dados, vamos entrar no código com a linguagem R!

Implementando Regras de Associação usando R

O Conjunto de Dados

Neste laboratório, você usará um conjunto de dados do repositório da UCI, que pode ser acessado em <http://archive.ics.uci.edu/ml/index.php>.

O repositório de aprendizado de máquina da UCI é uma coleção de bancos de dados, teorias de domínio e geradores que são usados pela comunidade de aprendizado de máquina para a análise empírica de algoritmos de aprendizado de máquina. O repositório foi criado como um arquivo FTP em 1987 por David Aha e outros estudantes de pós-graduação na UC Irvine. Desde aquela época, tem sido amplamente utilizado por estudantes, educadores e pesquisadores de todo o mundo como uma fonte primária de conjuntos de dados de aprendizado de máquina. Como uma indicação do impacto deste repositório, ele foi citado mais de 1000 vezes, tornando-se um dos Top 100 mais citados "papers" em toda a ciência da computação. A versão atual do Web Site foi projetada em 2007 por Arthur Asuncion e David Newman, e este projeto é realizado em colaboração com [Rexa.info](http://www.rexa.info) na Universidade de Massachusetts Amherst, com apoio financeiro da National Science Foundation. Muitas pessoas merecem agradecimentos por tornar este repositório um sucesso. Dentre eles, estão os doadores e criadores dos bancos de dados e geradores.

O conjunto de dados é chamado **Online-Retail** (varejo-online), e você pode baixá-lo deste endereço: <http://archive.ics.uci.edu/ml/datasets/online+retail>. Este dataset contém dados de transações de 01/12/2010 a 09/12/2011 de um varejo on-line registrado no Reino Unido. O motivo para usar este e não o conjunto de dados exemplo disponível no R é que é mais provável que você receba informações de varejo neste formato, no qual você terá que aplicar um pré-processamento ou tratamento de dados para poder utilizá-lo no R.

Descrição do Dataset

- Número de Linhas: 541.909
- Número de Atributos: 8

Informações dos Atributos

- **InvoiceNo:** Número da fatura. Nominal, um número inteiro de 6 dígitos atribuído exclusivamente a cada transação. Se este número começar com letra 'c', indicará um cancelamento.
- **StockCode:** Código do produto (item). Nominal, um número inteiro de 5 dígitos atribuído exclusivamente a cada produto distinto.
- **Description:** nome do produto (item). Nominal.
- **Quantity:** as quantidades de cada produto (item) por transação. Numérico.
- **InvoiceDate:** data e hora da fatura. Numérico, o dia e a hora em que cada transação foi gerada. Exemplo do Dataset: 12/1/2010 8:26
- **UnitPrice:** Preço unitário. Numérico, preço do produto por unidade.
- **Customerid:** Número do cliente. Nominal, um número inteiro de 5 dígitos atribuído exclusivamente a cada cliente.
- **Country:** Nome do país. Nominal, o nome do país onde cada cliente reside.

Carregando Bibliotecas do R

Inicialmente você carregará as bibliotecas R necessárias para a execução deste laboratório. Uma breve descrição das bibliotecas, para que você saiba o que cada biblioteca faz, é apresentada na tabela a seguir:

Pacote	Descrição
arules	Fornecer a infraestrutura para representar, manipular e analisar dados de transação e padrões (conjuntos de itens frequentes e regras de associação).
arulesViz	Estende o pacote 'arules' com várias técnicas de visualização para regras de associação e conjuntos de itens. O pacote também inclui várias visualizações interativas para exploração de regras.
tidyverse	Uma coleção opinativa de pacotes de R projetados para a ciência de dados
readxl	Permite ler arquivos do Excel em R
plyr	Ferramentas para dividir, aplicar e combinar dados
ggplot2	Criar gráficos
knitr	Geração de relatórios dinâmicos em R
lubridate	Pacote R que facilita o trabalho com datas e horas.

```
#instalar e carregar o pacote arules
install.packages("arules")
library(arules)
#instalar e carregar arulesViz
install.packages("arulesViz")
library(arulesViz)
#instalar e carregar tidyverse
install.packages("tidyverse")
library(tidyverse)
#instalar e carregar Readxl
install.packages("readxl")
library(readxl)
#instalar e carregar o knitr
install.packages("knitr")
library(knitr)
#Carregar o ggplot2 pois ele já vem em tidyverse
library(ggplot2)
#instalar e carregar o lubridate
install.packages("lubridate")
library(lubridate)
#instalar e carregar plyr
install.packages("plyr")
library(plyr)
library(dplyr)

# UFA!!! Quantos pacotes e bibliotecas!! Mas o R é assim mesmo!!
```

Pré-processamento de dados

Usar `read_excel(caminho para o arquivo)` para ler para o R o conjunto de dados do arquivo de dados das transações. Dê o caminho completo para arquivo, incluindo o nome do arquivo: `read_excel(caminho-do-arquivo-com-nome-do-arquivo)`

```
#Ler os dados do Excel para o R dataframe
library(readxl)
retail <- read_excel('c:/dados/Online_Retail.xlsx')
view(retail)
```

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
1	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850	United Kingdom
2	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
3	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850	United Kingdom
4	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
5	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850	United Kingdom
6	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850	United Kingdom
7	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850	United Kingdom
8	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
9	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850	United Kingdom
10	536367	84679	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047	United Kingdom
11	536367	22745	POPPY'S PLAYHOUSE BEDROOM	6	2010-12-01 08:34:00	2.10	13047	United Kingdom
12	536367	22748	POPPY'S PLAYHOUSE KITCHEN	6	2010-12-01 08:34:00	2.10	13047	United Kingdom

Showing 1 to 14 of 541,909 entries, 8 total columns

Dados de transações de varejo lidos do arquivo Excel

```
#complete.cases(data) retornará um vetor lógico que indica quais linhas não têm valores
#ausentes. Em seguida, use o vetor para obter apenas as linhas que estão completas
#usando o comando a seguir:
retail <- retail[complete.cases(retail), ]
#mutate esta função é do pacote dplyr. Ela é usado para editar ou adicionar novas
#colunas para dataframes. Aqui a coluna description está sendo convertida para a coluna
#de fator. as.factor converte a coluna para fator coluna. %>% é um operador com o qual
#você pode canalizar valores para outra função ou expressão
retail %>% mutate(Description = as.factor(Description))
retail %>% mutate(Country = as.factor(Country))
#Converte dados de caracteres para data. Converte InvoiceDate como data na nova variável
retail$Date <- as.Date(retail$InvoiceDate)
#Extraí a hora de InvoiceDate e armazena em outra variável
TransTime<- format(retail$InvoiceDate,"%H:%M:%S")
#Converte e edita InvoiceNo em numérico
InvoiceNo <- as.numeric(as.character(retail$InvoiceNo))
#NAs introduzido por coerção
#Acrescenta as novas colunas TransTime e InvoiceNo no dataframe retail
cbind(retail,TransTime)
cbind(retail,InvoiceNo)
# Finalmente, dê uma olhada agora nos seus dados
glimpse(retail)
```

```
Observations: 406,829
Variables: 9
$ InvoiceNo    <chr> "536365", "536365", "536365", "536365", "536365", "536365", "53...
$ StockCode   <chr> "85123A", "71053", "84406B", "84029G", "84029E", "22752", "2173...
$ Description  <chr> "WHITE HANGING HEART T-LIGHT HOLDER", "WHITE METAL LANTERN", "C...
$ Quantity    <dbl> 6, 6, 8, 6, 6, 2, 6, 6, 6, 32, 6, 6, 8, 6, 6, 3, 2, 3, 3, 4, 4,...
$ InvoiceDate  <dtm> 2010-12-01 08:26:00, 2010-12-01 08:26:00, 2010-12-01 08:26:00,...
$ UnitPrice   <dbl> 2.55, 3.39, 2.75, 3.39, 3.39, 7.65, 4.25, 1.85, 1.85, 1.69, 2.1...
$ CustomerID  <dbl> 17850, 17850, 17850, 17850, 17850, 17850, 17850, 17850, 17850, ...
$ Country     <chr> "United Kingdom", "United Kingdom", "United Kingdom", "United K...
$ Date        <date> 2010-12-01, 2010-12-01, 2010-12-01, 2010-12-01, 2010-12-01, 20...
```

Agora o dataframe **retail** conterá 10 atributos, com dois atributos adicionais Date e Time.

Antes de aplicar o algoritmo MBA/AssociationRule de mineração de dados, precisamos converter o *dataframe* em dados de transação para que todos os itens comprados juntos em uma fatura estejam em uma linha. Você pode ver na saída acima que cada transação está em forma atômica, isto é, todos os produtos pertencentes a uma fatura são dados individuais, como em bancos de dados relacionais. Este formato também é chamado como **singles**.

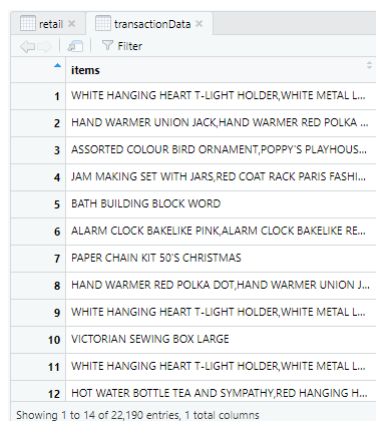
O que você precisa fazer é agrupar dados no dataframe **retail** quer por Customerid, Customerid e Date ou você também pode agrupar dados usando InvoiceNo e Date. Precisamos deste agrupamento para executar o algoritmo de associação. Vamos aplicar uma função no conjunto de dados retail e armazenar a saída em outro dataframe. Isso pode ser feito por com a função **ddply**.

As linhas de código a seguir combinarão todos os produtos de um InvoiceNo e Date e combinará todos os produtos de uma InvoiceNo e Date como uma fileira, com cada item, separados por vírgula.

```
library(plyr)
# ddply(dataframe, variables_to_be_used_to_split_data_frame, function_to_be_applied)
transactionData <- ddply(retail, c("InvoiceNo", "Date"),
function(df1) paste(df1$Description, collapse = ","))
# A função R paste() concatena vetores para caracteres e resultados separados usando
# collapse=[any optional charcater string ]. Aqui ', ' é usado.
transactionData
```

Em seguida, como InvoiceNo e Date não terão qualquer uso na aplicação das regras de associação, você pode defini-los para Null.

```
#set coluna InvoiceNo do dataframe transactionData
transactionData$InvoiceNo <- NULL
#set coluna Date de dataframe transactionData
transactionData$Date <- NULL
#Renomeie a coluna para itens
colnames(transactionData) <- c("items")
#Mostre agora novamente o dataframe transactionData
transactionData
View(transactionData)
```



	items
1	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL L...
2	HAND WARMER UNION JACK,HAND WARMER RED POLKA ...
3	ASSORTED COLOUR BIRD ORNAMENT,POPPY'S PLAYHOU...
4	JAM MAKING SET WITH JARS,RED COAT RACK PARIS FASHI...
5	BATH BUILDING BLOCK WORD
6	ALARM CLOCK BAKELIKE PINK,ALARM CLOCK BAKELIKE RE...
7	PAPER CHAIN KIT 50'S CHRISTMAS
8	HAND WARMER RED POLKA DOT,HAND WARMER UNION J...
9	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL L...
10	VICTORIAN SEWING BOX LARGE
11	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL L...
12	HOT WATER BOTTLE TEA AND SYMPATHY,RED HANGING H...

Showing 1 to 14 of 22,190 entries, 1 total columns

Esse formato para dados de transação é chamado de Formato **Basket**. Em seguida, você deve armazenar esses dados de transação em um arquivo **.csv** (valores separados por vírgula). Para isso, utilize a função **write.csv()**.

```
write.csv(transactionData,"C:/dados/market_basket_transactions.csv", quote = FALSE,
row.names = FALSE)
#transactionData: Dados a serem gravados
#"C:/dados/market_basket.csv": localização e nome do arquivo a ser gravado
#quote: Se TRUE ele irá delimitar a coluna de caractere ou fator com aspas duplas.
#Se False nada será feito
#row.names: um valor lógico indicando se os nomes de linha de x devem ser gravados junto
#com x, ou um vetor de caracteres de nomes de linha a serem gravados.
```

Utilizando o Excel, veja se seus dados de transação, que acabaram de ser gerados para o arquivo **c:/dados/market_basket_transactions.csv**, estão com a forma correta conforme ilustrado a seguir:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	Items																												
2	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL LANTERN,CREAM CUPID HEARTS COAT HANGER,KNITTED UNION FLAG HOT WATER BOTTLE,RED WOOLLY HOTTIE WHITE HEART,,SET 7 BABUSHKA NESTING BOXES,GLASS STAR FROSTED T-LIGHT HOLDER																												
3	HAND WARMER UNION JACK,HAND WARMER RED POLKA DOT																												
4	ASSORTED COLOUR BIRD ORNAMENT,POPPY'S PLAYHOUSE KITCHEN,FELTCRAFT PRINCESS CHARLOTTE DOLL,IVORY KNITTED MUG COSY,BOX OF 6 ASSORTED COLOUR TEASPOONS,BOX OF VINTAGE JIGSAW BLOCKS,BOX OF VINTAGE ALPHABET BLOCKS,HOME BUILDING BLOCK W																												
5	JAM MAKING SET WITH JARS,RED COAT RACK PARIS FASHION,YELLOW COAT RACK PARIS FASHION,BLUE COAT RACK PARIS FASHION																												
6	BATH BUILDING BLOCK WORD																												
7	ALARM CLOCK BAKELIKE PINK,ALARM CLOCK BAKELIKE GREEN,PANDA AND BUNNIES STICKER SHEET,STARS GIFT TAPE,INFLATABLE POLITICAL GLOBE,VINTAGE HEADS AND TAILS CARD GAME,SET/2 RED RETROSPOT TEA TOWELS,ROUND SNACK BOXES SET OF 4 WOODLAND,SPACEBO																												
8	PAPER CHAIN KIT 50'S CHRISTMAS																												
9	HAND WARMER RED POLKA DOT,HAND WARMER UNION JACK																												
10	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL LANTERN,CREAM CUPID HEARTS COAT HANGER,EDWARDIAN PARASOL,RED,RETRO COFFEE MUGS ASSORTED,SAVE THE PLANET MUG,VINTAGE BILLBOARD DRINK ME MUG,VINTAGE BILLBOARD LOVE/HATE MUG,WOOD 2 DRAWER CABINET WHITE FINISH,V																												
11	VICTORIAN SEWING BOX LARGE																												
12	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL LANTERN,CREAM CUPID HEARTS COAT HANGER,EDWARDIAN PARASOL,RED,RETRO COFFEE MUGS ASSORTED,SAVE THE PLANET MUG,VINTAGE BILLBOARD DRINK ME MUG,VINTAGE BILLBOARD LOVE/HATE MUG,WOOD 2 DRAWER CABINET WHITE FINISH,V																												
13	HOT WATER BOTTLE TEA AND SYMPATHY,RED HANGING HEART T-LIGHT HOLDER																												
14	HAND WARMER RED POLKA DOT,HAND WARMER UNION JACK																												
15	JUMBO BAG PINK POLKADOT,JUMBO BAG BAROQUE BLACK WHITE,JUMBO BAG CHARLIE AND LOLA TOYS,STRAWBERRY CHARLOTTE BAG,RED 3 PIECE RETROSPOT CUTLERY SET,BLUE 3 PIECE POLKADOT CUTLERY SET,SET/6 RED SPOTTY PAPER PLATES,LUNCH BAG RED RETROSPOT,STRAWBERRY LUNCH BOX WITH C																												
16	JAM MAKING SET PRINTED																												
17	RETROSPOT TEA SET CERAMIC 11 PC,GIRLY PINK TOOL SET,JUMBO SHOPPER VINTAGE RED FAISLEY,AIRLINE LOUNGE,METAL SIGN,WHITE SPOT RED CERAMIC DRAWER KNOB,RED CERAMIC DRAWER KNOB,ACRYLIC EDWARDIAN,CLEAR DRAWER KNOB,ACRYLIC EDWARDIAN,PHOTO CLIP LINE,FELT EGG COSY CHICKEN,PIGTOY																												
18	INFLATABLE POLITICAL GLOBE,VINTAGE SNAKE & LADDERS,CHOCOLATE CALCULATOR,JUMBO SHOPPER VINTAGE RED FAISLEY,RECYCLING BAG RETROSPOT,TOY TIEP PINK POLKADOT,ANTIQUE GLASS DRESSING TABLE,POT,ALARM CLOCK BAKELIKE GREEN,IVORY GIANT GARDEN THERMOMETER,3 TIER CAKE TIN																												
19	WOOD BLACK BOARD ANT WHITE FINISH,COLOUR GLASS T-LIGHT HOLDER,HANGING,HANGING METAL HEART LANTERN,HANGING MEDINA LANTERN SMALL,NATURAL SLATE HEART CHALKBOARD,HEART OF WICKER SMALL,HEART OF WICKER LARGE,WHITE LOVEBIRD LANTERN,CLASSIC METAL BIRDCAVE PLANT H																												
20	SET 3 WICKER OVAL BASKETS W LIDS,JAM MAKING SET PRINTED,JAM MAKING SET WITH JARS,JUMBO BAG DOLLY GIRL DESIGN,TRADITIONAL CHRISTMAS RIBBONS,ORGANISER WOOD ANTIQUE WHITE,LUNCH BAG DOLLY GIRL DESIGN																												
21	WHITE WIRE EGG HOLDER,JUMBO BAG BAROQUE BLACK WHITE,JUMBO BAG RED RETROSPOT																												
22	CHILLI LIGHTS,LIGHT GARLAND BUTTERFLIES PINK,WOODEN OWLS LIGHT GARLAND FAIRY TALE COTTAGE NIGHTLIGHT,RED TOADSPOOL LED NIGHT LIGHT																												
23	HOME BUILDING BLOCK WORD,LOVE BUILDING BLOCK WORD,DOORMAT FANCY FONT HOME SWEET HOME,HOME SMALL WOOD LETTERS,GINGHAM HEART DOORSTOP,RED,FIVE HEART HANGING DECORATION,HANGING METAL HEART LANTERN,ASSORTED BOTTLE TOP MAGNETS,FRIDGE MAGNETS US DINER AT																												
24	CHRISTMAS LIGHTS 10 REINDEER,VINTAGE UNION JACK CUSHION COVER,VINTAGE HEADS AND TAILS CARD GAME,SET OF 3 COLOURED FLYING DUCKS,SET OF 3 GOLD FLYING DUCKS,RED RETROSPOT UMBRELLA,BLACK/BLUE POLKADOT UMBRELLA,RED DINER WALL CLOCK,ALARM CLOCK BAKELIKE GREEN,ALARM																												
25	CHRISTMAS LIGHTS 10 REINDEER,JAM MAKING SET WITH JARS,JAM MAKING SET PRINTED,JAM JAR WITH PINK LID,JAM JAR WITH GREEN LID,ROSE COTTAGE KEEPSAKE BOX,HANGING HEART ZINC T-LIGHT HOLDER,PAPER CHAIN KIT VINTAGE CHRISTMAS,DISCO BALL CHRISTMAS DECORATION,WHITE HANGING HE																												
26	3 STRIPY YACE FELTCRAFT,SET OF 5 SOLDIER SITTLES,TRADITIONAL WOODEN SHIPPING ROPE,WOODEN BOX OF DOMINOES,RUSTIC SEVENTEEN DRAWER SIDEBORD,PARTY CONES CARNIVAL ASSORTED,PARTY CONES CANDY ASSORTED,PONC BASKET WICKER SMALL,ASSORTED COLOUR BIRD ORNAMENT,ST																												
27	RETROSPOT LAMP																												
28	FANCY FONT BIRTHDAY CARD,,HAND WARMER UNION JACK,HAND WARMER SCOTTY DOG DESIGN,HAND WARMER OWL DESIGN,HAND WARMER RED RETROSPOT,RETROSPOT HEART HOT WATER BOTTLE,DOG BOWL CHASING BALL DESIGN,CLOTHES PEGS RETROSPOT PACK 24,HAND OVER THE CHOCOLATE SIGN																												
29	BLACK HEART CARD HOLDER,ASSORTED COLOUR BIRD ORNAMENT,PACK OF 60 PINK FAISLEY CAKE CASES,60 TEATIME FAIRY CAKE CASES,PACK OF 72 RETROSPOT CAKE CASES,CHICK GREY HOT WATER BOTTLE,SMALL GLASS HEART TRINKET POT,ALARM CLOCK BAKELIKE IVORY,ALARM CLOCK BAKELIKE RED,ALARM																												
30	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL LANTERN,CREAM CUPID HEARTS COAT HANGER,EDWARDIAN PARASOL,RED,RETRO COFFEE MUGS ASSORTED,SAVE THE PLANET MUG,VINTAGE BILLBOARD DRINK ME MUG,VINTAGE BILLBOARD LOVE/HATE MUG,WOOD 2 DRA																												
31	SET OF 3 BLACK FLYING DUCKS,SET OF 3 COLOURED FLYING DUCKS																												
32	PACK OF 12 RED RETROSPOT TISSUES,RED RETROSPOT MUG,BABUSHKA LIGHTS STRING OF 10,PIGTOY BANK RETROSPOT,SET 7 BABUSHKA NESTING BOXES,DOORMAT FAIRY CAKE,HAND WARMER RED RETROSPOT,HAND WARMER SCOTTY DOG DESIGN,HAND WARMER OWL DESIGN,STRAWBERRY CERAMIC TRINKET																												
33	HAND WARMER RED POLKA DOT,HAND WARMER UNION JACK																												
34	HOMEMADE JAM SCENTED CANDLES																												
35	BIRD HOUSE HOT WATER BOTTLE,BOUDOIR SQUARE TISSUE BOX,SKULLS SQUARE TISSUE BOX,PHOTO FRAME CORNICE,SILK PURSE BABUSHKA RED,PICTURE DOMINOES,5/6 SEW ON CROCHET FLOWERS,SCANDINAVIAN REDS RIBBONS,BALLOONS WRITING SET,JAM MAKING SET PRINTED,LAVENDER INCENSE IN T																												
36	PAPER CHAIN KIT 50'S CHRISTMAS,PAPER CHAIN KIT VINTAGE CHRISTMAS,HOT WATER BOTTLE BABUSHKA																												
37	HAND WARMER BIRD DESIGN,POSTAGE																												
38	HEART IVORY TRELLIS SMALL,CLEAR DRAWER KNOB ACRYLIC EDWARDIAN,PINK DRAWER KNOB ACRYLIC EDWARDIAN,GREEN DRAWER KNOB ACRYLIC EDWARDIAN,BLUE DRAWER KNOB ACRYLIC EDWARDIAN,HEART OF WICKER SMALL,SMALL POPCORN HOLDER,SET 6 FOOTBALL CELEBRATION CANDLES,SET OF 6																												
39	SET/5 RED RETROSPOT LID GLASS BOWLS																												
40	WHITE HANGING HEART T-LIGHT HOLDER,WHITE METAL LANTERN,CREAM CUPID HEARTS COAT HANGER,EDWARDIAN PARASOL,RED,RETRO COFFEE MUGS ASSORTED,SAVE THE PLANET MUG,VINTAGE BILLBOARD DRINK ME MUG,VINTAGE BILLBOARD LOVE/HATE MUG,WOOD 2 DRAWER CABINET WHITE FINISH,V																												
41	HAND WARMER RED POLKA DOT,HAND WARMER UNION JACK																												
42	MAGIC DRAWING SLATE DINOSAUR,MAGIC DRAWING SLATE BAKE A CAKE,12 PENCILS TALL TUBE SKULLS,CHOCOLATE CALCULATOR,RED HARMONICA IN BOX,BLUE HARMONICA IN BOX,SKULLS WATER TRANSFER TATTOOS,PACK 3 BOXES BIRD PANNETONE,HOT WATER BOTTLE TEA AND SYMPATHY,RED WOOLLY																												
43	5 STRAND GLASS NECKLACE CRYSTAL,WHITE SKULL HOT WATER BOTTLE,SCOTTIE DOG HOT WATER BOTTLE,SQUARECUSHION COVER PINK UNION FLAG,SPACEBOY CHILDRENS EGG CUP,CHILDRENS SPACEBOY MUG,HAND WARMER SCOTTY DOG DESIGN,FELTCRAFT CUSHION OWL,BLACK CANDLEABRA T-LIGHT HO																												
44	LUNCH BAG CARLS BLUE,LUNCH BAG SPACEBOY DESIGN,ROUND SNACK BOXES SET OF 4 WOODLAND,LUNCH BAG DOLLY GIRL DESIGN,LUNCH BAG SUKI DESIGN,LUNCH BAG BLACK SKULL,,ROUND SNACK BOXES SET OF 4 FRUITS,ROUND SNACK BOXES SET OF 4 SKULLS,DOLLY GIRL LUNCH BOX,SPACEBOY LUNCH BO																												
45	40 CAKE CASES VINTAGE CHRISTMAS,PAPER CHAIN KIT VINTAGE CHRISTMAS,RIBBON REEL CHRISTMAS SOCK BAUBLE RIBBON REEL SNOWY VILLAGE,RIBBON REEL MAKING SNOWMEN,SET OF 20 VINTAGE CHRISTMAS NAPKINS,TURQUOISE CHRISTMAS TREE,RED STAR CARD HOLDER,WICKER WREATH SMALL,HEA																												
46	ROTATING LEAVES T-LIGHT HOLDER,RED HARMONICA IN BOX,CUPCAKE LACE PAPER SET & FAMILY PHOTO FRAME CORNICE,TRIPLE PHOTO FRAME CORNICE,WOODEN FRAME ANTIQUE WHITE																												

Em seguida, você deve carregar esses dados de transação em um objeto da classe de transação. Isso é feito usando a função R **read.transactions** do pacote **arules**.

A linha de código apresentada a seguir lerá o arquivo de dados de transação, que está em no formato **Basket .csv**, **C:/dados/market_basket_transactions.csv**, e o converterá para um objeto da classe de transação, que denominaremos simplesmente **tr**.

```
library(arules)
tr <- read.transactions('C:/Dados/market_basket_transactions.csv', format = 'basket',
sep=',')
#sep diz como os itens são separados. Neste caso, você separou usando vírgulas ','
```

Quando você executa as linhas de código acima pode obter muitos EOF dentro da cadeia de caracteres da sua saída, o que fará com que o R relate alguns “warnings”, mas não se preocupe com isso.

Quando você já tiver dados de transação em um dataframe, que não foi o nosso caso, poderá utilizar a seguinte linha de código para convertê-lo em um objeto de transação:

```
trObj<-as(dataframe.dat,"transactions")`
```

Veja o agora o seu **tr** (objeto de transação) que foi criado:

```
tr
transactions in sparse format with
  22191 transactions (rows) and
  30066 items (columns)
summary(tr)
```

```
transactions in sparse format with
  22191 transactions (rows) and
  7876 items (columns)
transactions as itemMatrix in sparse format with
  22191 rows (elements/itemsets/transactions) and
  7876 columns (items) and a density of 0.001930725
```

most frequent items:

WHITE HANGING HEART T-LIGHT HOLDER	1803	REGENCY CAKESTAND 3 TIER	1709
JUMBO BAG RED RETROSPOT	1460	PARTY BUNTING	1285
ASSORTED COLOUR BIRD ORNAMENT	1250	(Other)	329938

```
element (itemset/transaction) length distribution:
sizes
```

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----


```

3598 1594 1141 908 861 758 696 676 663 593 624 537 516 531 551 522 464
18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
441 483 419 395 315 306 272 238 253 229 213 222 215 170 159 138 142
35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51
134 109 111 90 113 94 93 87 88 65 63 67 63 60 59 49 64
52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68
40 41 49 43 36 29 39 30 27 28 17 25 25 20 27 24 22
69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85
15 20 19 13 16 16 11 15 12 7 9 14 15 12 8 9 11
86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102
11 14 8 6 5 6 11 6 4 4 3 6 5 2 4 2 4
103 104 105 106 107 108 109 110 111 112 113 114 116 117 118 120 121
4 3 2 2 6 3 4 3 2 1 3 1 3 3 3 1 2
122 123 125 126 127 131 132 133 134 140 141 142 143 145 146 147 150
2 1 3 2 2 1 1 2 1 2 2 1 1 2 1 1
154 157 168 171 177 178 180 202 204 228 236 249 250 285 320 400 419
3 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1

Min. 1st Qu. Median Mean 3rd Qu. Max.
1.00 3.00 10.00 15.21 21.00 419.00

includes extended item information - examples:
labels
1 1 HANGER
2 10 COLOUR SPACEBOY PEN
3 12 COLOURED PARTY BALLOONS

```

O `summary(tr)` é um comando muito útil que nos dá informações sobre o nosso objeto de transação. Vamos dar uma olhada no que a saída acima mostra:

- Existem **22.191 transações (linhas) e 7.876 itens (colunas)**. Observe que **7.876** é a descrição do produto envolvida no conjunto de dados e **22.191** transações são coleções desses itens.
- **Density** informa a porcentagem de células não-zero em uma matriz esparsa. Você pode considerar como o número total de itens que são comprados divididos por um número possível de itens nessa matriz. Você pode calcular quantos itens foram comprados usando densidade: $22191 \times 7876 \times 0.001930725 = 337445$.

Matriz Esparsa: uma matriz esparsa é uma matriz na qual a maioria dos elementos são zero. Por outro lado, se a maioria dos elementos são diferente de zero, então, a matriz é considerada densa. O número de elementos com valor zero, dividido pelo número total de elementos, é chamado de dispersão da matriz (que é igual a 1 menos a densidade da matriz).

- **Summary** também pode dizer-lhe os itens mais frequentes.
- **Element (itemset/transaction) length distribution:** mostra quantas transações estão lá para 1-Itemset, para 2-Itemset e assim por diante. A primeira linha mostra um número de Itens enquanto a segunda linha mostra o número de transações.

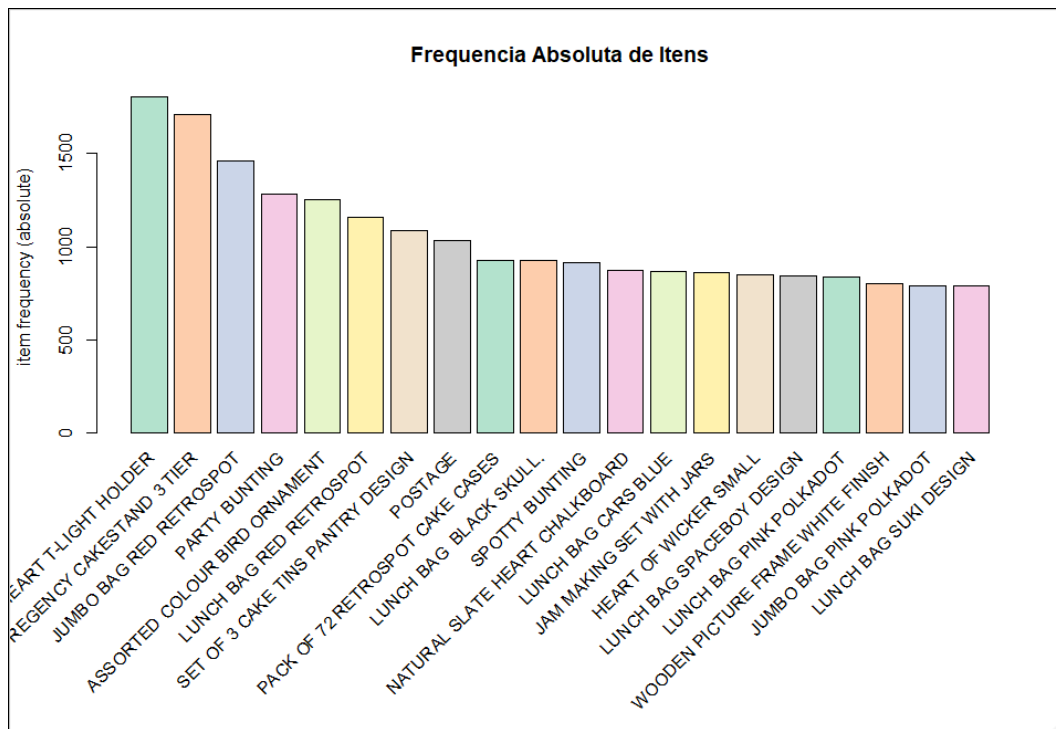
Por exemplo, há apenas 3.598 transações para 1 item, 1.594 transações para 2 itens, e há 419 itens em uma transação que é a mais longa (Max.).

Você pode gerar um gráfico **itemFrequencyPlot** para criar um gráfico de barras de frequência para exibir a distribuição de objetos com base em **itemMatrix** (por exemplo, > transações ou itens em > itemsets e > regras) que é o nosso caso.

```

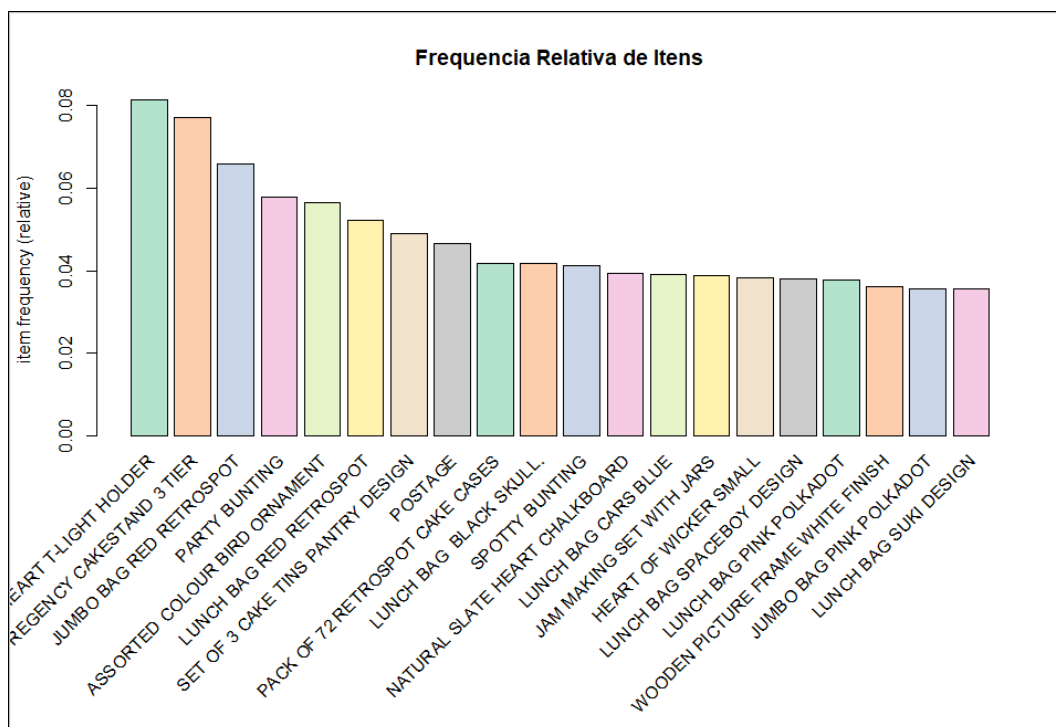
# Criar um gráfico de frequência de item para os 20 itens principais
if (!require("RColorBrewer")) {
  # install color package of R
  install.packages("RColorBrewer")
  #include library RColorBrewer
  library(RColorBrewer)
}
itemFrequencyPlot(tr,topN=20,type="absolute",col=brewer.pal(8,'Pastel12'),
main="Frequencia Absoluta de Itens")

```



Em `itemFrequencyPlot(tr,topN=20,type="absolute")` o primeiro argumento é o objeto de transação a ser plotado que é **tr**. **topN** permite que você plote os primeiros N itens de maior frequência. **Type** pode ser “absolute” ou “relative”. Se absoluto serão plotadas frequências numéricas de cada item de forma independente. Se relativo serão plotadas quantas vezes esses itens apareceram em comparação com outros.

```
itemFrequencyPlot(tr,topN=20,type="relative",col=brewer.pal(8,'Pastel2'),main="Frequencia Relativa de Itens")
```



Este gráfico mostra que os primeiros produtos (itens) 'WHITE HANGING HEART T-LIGHT HOLDER' e 'REGENCY CAKESTAND 3 TIER' são responsáveis pela maioria das vendas. Então para aumentar a venda

do produto (item) 'SET OF 3 CAKE TINS PANTRY DESIGN ' o varejista pode colocá-lo perto do produto (item) 'REGENCY CAKESTAND 3 TIER'.

Gerando Regras

O próximo passo é gerar as regras usando o algoritmo APRIORI. A função `Apriori()` está contida no pacote `arules`.

```
# Min suporte como 0.001, confiança como 0,8.
association.rules <- apriori(tr, parameter = list(supp=0.001, conf=0.8,maxlen=10))

Apriori
Parameter specification:
 confidence minval smax arem aval originalSupport maxtime support minlen maxlen target
           0.8    0.1    1 none FALSE                TRUE      5  0.001      1    10 rules
  ext
FALSE

Algorithmic control:
 filter tree heap memopt load sort verbose
    0.1 TRUE TRUE  FALSE TRUE     2     TRUE

Absolute minimum support count: 22

set item appearances ...[0 item(s)] done [0.00s].
set transactions ...[30066 item(s), 22191 transaction(s)] done [0.11s].
sorting and recoding items ... [2324 item(s)] done [0.02s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10
Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!
done [0.70s].
writing ... [49122 rule(s)] done [0.06s].
creating S4 object ... done [0.06s].
set of 49122 rules

rule length distribution (lhs + rhs):sizes
   2    3    4    5    6    7    8    9   10
 105 2111 6854 16424 14855 6102 1937 613 121

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 2.000   5.000   5.000   5.499   6.000  10.000

summary of quality measures:
      support      confidence      lift      count
Min. :0.001036  Min. :0.8000  Min. : 9.846  Min. : 23.00
1st Qu.:0.001082 1st Qu.:0.8333 1st Qu.: 22.237 1st Qu.: 24.00
Median :0.001262 Median :0.8788 Median : 28.760 Median : 28.00
Mean   :0.001417 Mean   :0.8849 Mean   : 64.589 Mean   : 31.45
3rd Qu.:0.001532 3rd Qu.:0.9259 3rd Qu.: 69.200 3rd Qu.: 34.00
Max.   :0.015997 Max.   :1.0000 Max.   :715.839 Max.   :355.00

mining info:
 data ntransactions support confidence
  tr          22191    0.001      0.8
```

O algoritmo Apriori tomará `tr` como o objeto de transação no qual a mineração deve ser aplicada. **parameter** permitirá que você defina `Min_sup` e `min_confidence`. Os valores padrão para este parâmetro são o suporte mínimo de 0,1, a confiança mínima de 0,8 e máximo de 10 itens (`maxlen`).

```
summary(association.rules)
```

Mostra o seguinte:

- **Especificação do parâmetro:** `min_sup= 0.001` e `min_confidence= 0.8` valores com 10 itens como máximo de itens em uma regra.
- **Número total de regras:** O conjunto de 49.122 Regras
- **Distribuição do comprimento da regra:** Um comprimento de 5 itens corresponde à maioria das regras: 16.424; e o comprimento de 2 itens têm o menor número de regras: 105.
- **Resumo das medidas de qualidade:** valores mínimos e máximos para suporte, confiança e Lift.

- **Informações usadas para criar regras:** Os dados, suporte e confiança que fornecemos ao algoritmo.

Uma vez que há **49.122** regras, vamos imprimir apenas as primeiras 10:

```
inspect(association.rules[1:10])
```

	lhs	rhs	support	confidence	lift	count
[1]	{WOBBLY CHICKEN}	=> {METAL}	0.001261773	1	443.82000	28
[2]	{WOBBLY CHICKEN}	=> {DECORATION}	0.001261773	1	443.82000	28
[3]	{DECOUPAGE}	=> {GREETING CARD}	0.001036456	1	389.31579	23
[4]	{BILLBOARD FONTS DESIGN}	=> {WRAP}	0.001306836	1	715.83871	29
[5]	{WOBBLY RABBIT}	=> {METAL}	0.001532153	1	443.82000	34
[6]	{WOBBLY RABBIT}	=> {DECORATION}	0.001532153	1	443.82000	34
[7]	{FUNK MONKEY}	=> {ART LIGHTS}	0.001712406	1	583.97368	38
[8]	{ART LIGHTS}	=> {FUNK MONKEY}	0.001712406	1	583.97368	38
[9]	{BLACK TEA}	=> {SUGAR JARS}	0.002072912	1	238.61290	46
[10]	{BLACK TEA}	=> {COFFEE}	0.002072912	1	69.34687	46

Usando a saída acima, você pode fazer a análises como:

- 100% dos clientes que compraram ' WOBBLY CHICKEN ' também compraram ' METAL '.
- 100% dos clientes que compraram ' BLACK TEA ' também compraram SUGAR ' JARS '.

Limitar o número e o tamanho das regras

Como limitar o tamanho e o número de regras geradas?

Você pode fazer isso definindo parâmetros no algoritmo Apriori. Você define esses parâmetros para ajustar o número de regras que receberá. Se você quiser regras mais fortes, pode aumentar o valor de **conf** e para regras mais estendidas dar um valor maior para **maxlen**. Por exemplo:

```
shorter.association.rules <- apriori(tr, parameter = list(supp=0.001,
conf=0.8,maxlen=3))
inspect(shorter.association.rules[1:10])
```

Removendo Regras Redundantes

Você pode remover regras que são subconjuntos de regras maiores. Use o código abaixo para remover essas regras (não se preocupe, demorará um pouco pois temos muitas regras):

```
subset.rules <- which(colSums(is.subset(association.rules, association.rules)) > 1)
# Obtém um subconjunto de regras em um vetor
length(subset.rules) #> 3913
[1] 44014
subset.association.rules. <- association.rules[-subset.rules]
# remove um subconjunto de regras
```

- **which()**: retorna a posição dos elementos no vetor para o qual o valor é TRUE.
- **colSums()**: forma uma linha e coluna somadas para dataframes e matrizes numéricas.
- **is.subset()**: determina se os elementos de um vetor contêm todos os elementos de outros

Encontrando Regras Relacionadas a Itens Fornecidos

Às vezes, você quer trabalhar em um produto específico. Se você quiser descobrir o que influência na compra do item **X** pode usar a opção **appearance** no comando **apriori**. Aparência nos dá opções para definir **LHS**(parte SE) e **RHS**(parte ENTÃO) da regra.

Por exemplo, para encontrar o que os clientes comprem antes de comprar ' METAL ' você pode executar a seguinte linha de código:

```
metal.association.rules <- apriori(tr, parameter = list(supp=0.001, conf=0.8),appearance
= list(default="lhs",rhs="METAL"))
```

```

Apriori

Parameter specification:
confidence minval smax arem aval originalSupport maxtime support minlen maxlen target
      0.8      0.1      1 none FALSE              TRUE      5   0.001      1     10 rules
  ext
FALSE

Algorithmic control:
filter tree heap memopt load sort verbose
  0.1 TRUE TRUE  FALSE TRUE     2     TRUE

Absolute minimum support count: 22

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[30066 item(s), 22191 transaction(s)] done [0.21s].
sorting and recoding items ... [2324 item(s)] done [0.02s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 3 4 5 6 7 8 9 10
Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!
done [0.63s].
writing ... [5 rule(s)] done [0.07s].
creating S4 object ... done [0.02s].

```

```

# Aqui LHS = METAL porque você quer descobrir as regras e probabilidades de
# Clientes que compram METAL junto com outros itens
inspect(head(metal.association.rules))

```

	lhs	rhs	support	confidence	lift	count
[1]	{WOBBLY CHICKEN}	=> {METAL}	0.001261773	1	443.82	28
[2]	{WOBBLY RABBIT}	=> {METAL}	0.001532153	1	443.82	34
[3]	{DECORATION}	=> {METAL}	0.002253166	1	443.82	50
[4]	{DECORATION,WOBBLY CHICKEN}	=> {METAL}	0.001261773	1	443.82	28
[5]	{DECORATION,WOBBLY RABBIT}	=> {METAL}	0.001532153	1	443.82	34

Da mesma forma, para encontrar a resposta para a pergunta Clientes que compraram METAL também compram.... Você vai manter METAL em lhs:

```

metal.association.rules <- apriori(tr, parameter = list(supp=0.001, conf=0.8),appearance
= list(lhs="METAL",default="rhs"))

Apriori

Parameter specification:
confidence minval smax arem aval originalSupport maxtime support minlen maxlen target
      0.8      0.1      1 none FALSE              TRUE      5   0.001      1     10 rules
  ext
FALSE

Algorithmic control:
filter tree heap memopt load sort verbose
  0.1 TRUE TRUE  FALSE TRUE     2     TRUE

Absolute minimum support count: 22

set item appearances ...[1 item(s)] done [0.00s].
set transactions ...[30066 item(s), 22191 transaction(s)] done [0.10s].
sorting and recoding items ... [2324 item(s)] done [0.02s].
creating transaction tree ... done [0.02s].
checking subsets of size 1 2 done [0.01s].
writing ... [1 rule(s)] done [0.00s].
creating S4 object ... done [0.01s].
# Here lhs=METAL because you want to find out the probability of that in how many
customers buy METAL along with other items
inspect(head(metal.association.rules))

```

	lhs	rhs	support	confidence	lift	count
[1]	{METAL}	=> {DECORATION}	0.002253166	1	443.82	50

Visualizando Regras de Associação

Uma vez que haverá centenas ou milhares de regras geradas com base nos dados de transações, você precisa de algumas maneiras melhores para apresentar suas descobertas. **ItemFrequencyPlot** já foi discutido acima e é uma grande alternativa de obter os produtos (itens) mais vendidos.

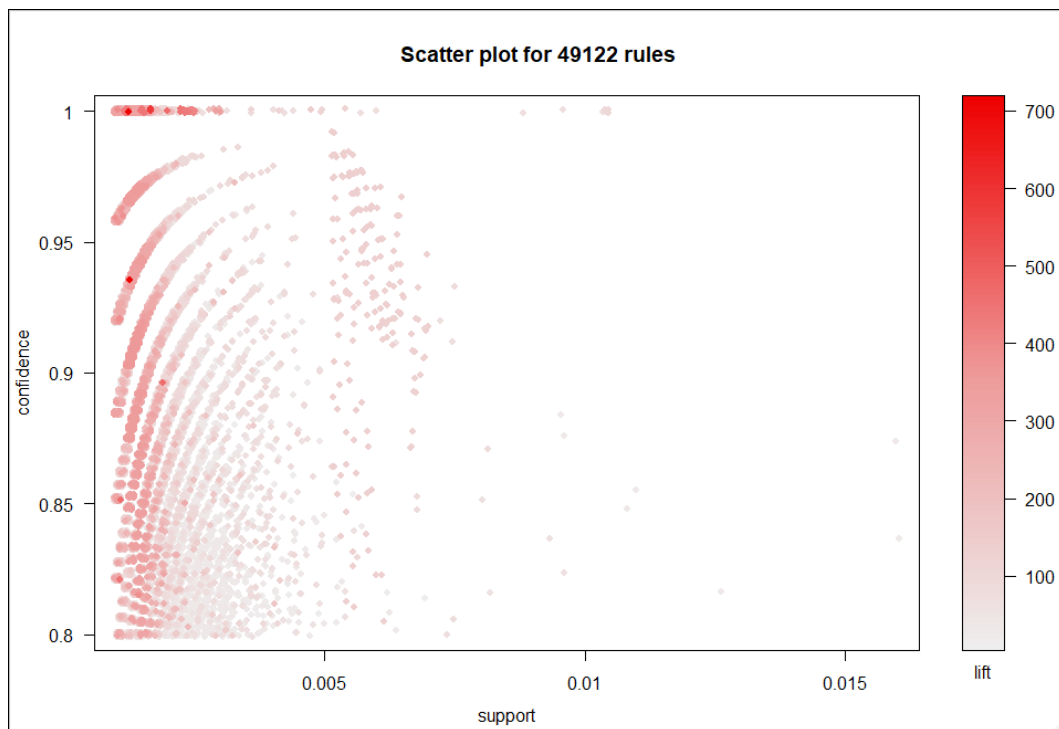
Aqui discutiremos as seguintes visualizações:

- Gráfico de dispersão
- Gráfico de dispersão interativo
- Representação individual de Regras

Gráfico de Dispersão

Uma possibilidade para visualização direta das regras de associação é gerar um gráfico de dispersão usando a função `plot()` do pacote **arulesViz**. Ela mostra *Suporte* e *Confiança* nos eixos. Além disso, a terceira medida *Lift* é usada por padrão para colorir (níveis de cor) os pontos.

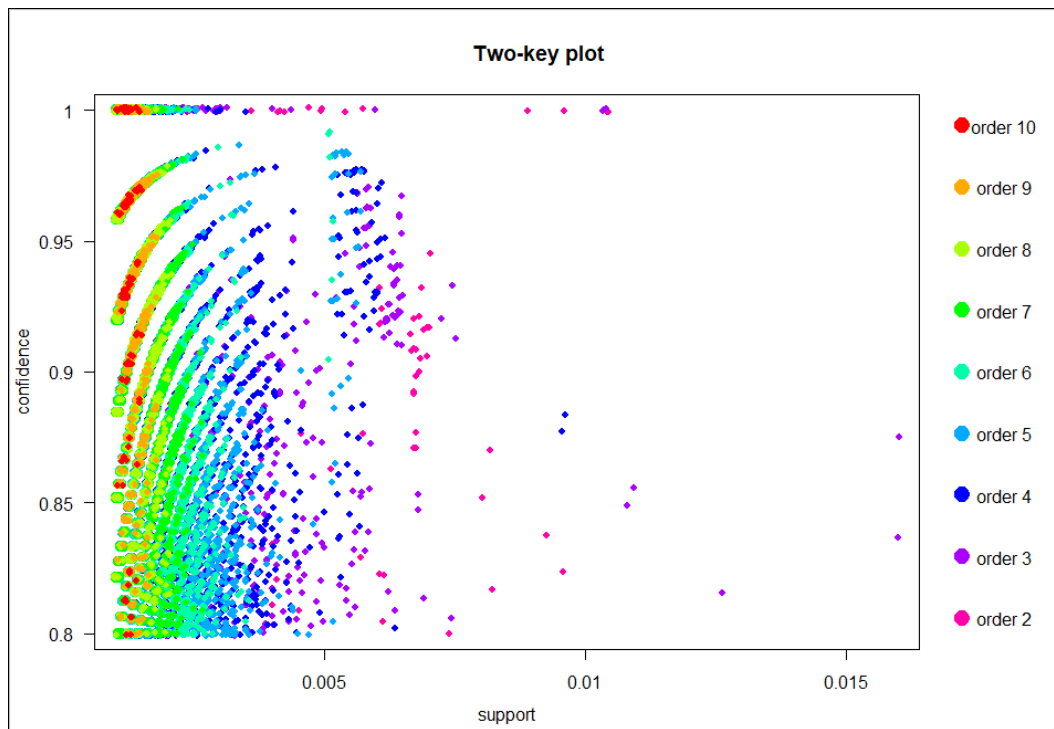
```
# Filtrar regras com confiança maior do que 0,4 ou 40%
library(arulesviz)
subRules<-association.rules[quality(association.rules)$confidence>0.4]
plot(subRules)
```



O gráfico mostra que as regras com alto **Lift** têm baixo **Suporte**. Você pode usar as seguintes opções para o gráfico: `plot(rulesObject, measure, shading, method)`, onde:

- **rulesObject**: o objeto de regras a ser plotado
- **measure**: medidas para o Interestingness da régua. Pode ser suporte, confiança, lift ou combinação destes dependendo do valor de `method`.
- **shading**: medida usada para colorir os pontos (suporte, confiança, lift). O padrão é Lift.
- **method**: método de visualização a ser usado (scatterplot, two-key plot, matrix3D).

```
plot(subRules,method="two-key plot")
```



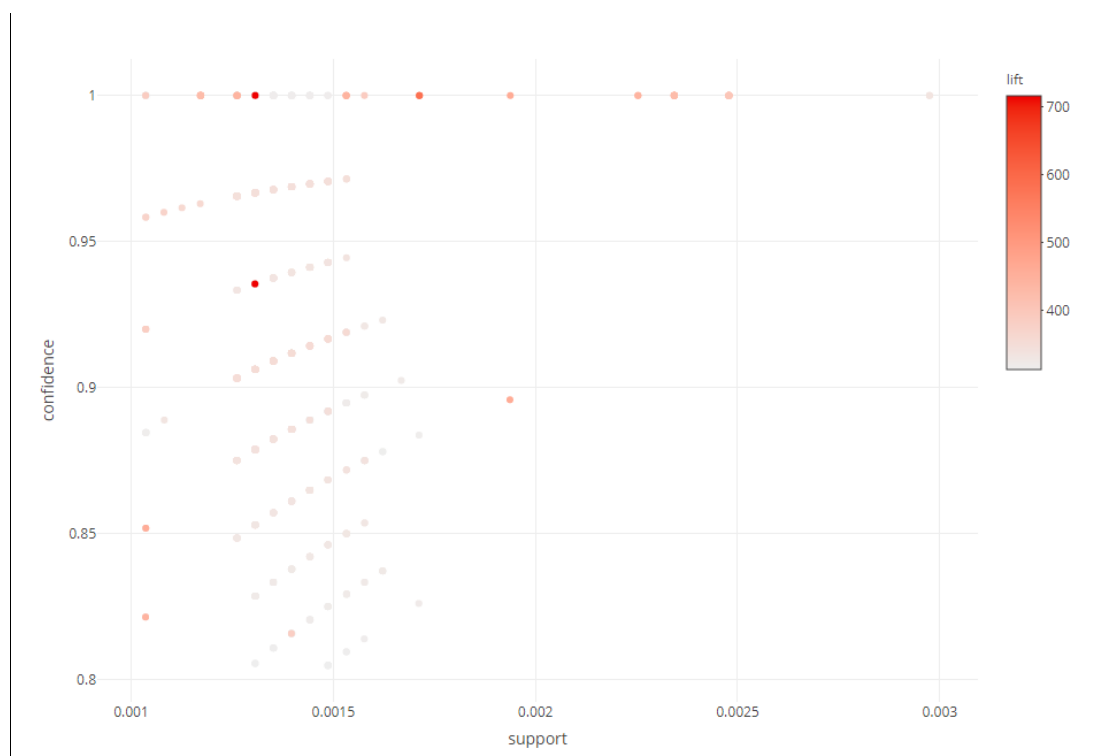
O gráfico **two-key** usa suporte e confiança nos eixos x e y respectivamente. Ele usa *order* (ordem) para colorir. A ordem é o número de itens na regra.

Gráfico de Dispersão Interativo

Um gráfico interativo pode ser gerado para apresentar suas regras que usam **arulesViz** e **plotly**. Nele você pode passar o mouse sobre cada regra e visualizar as medidas de qualidade (suporte, confiança e lift).

```
plotly_arules(subRules)
```

'plotly_arules' está obsoleto. Use 'plot' em vez disso.



Consulte Help ("preterido") Plot: Demasiadas regras fornecidas. Apenas plotando as melhores regras de 1000 usando o elevador de medida (mude o parâmetro Max se necessário) para Reduzir sobreplotagem, jitter é adicionado! Use jitter = 0 para evitar o jitter.

Visualizações Baseadas em Gráficos

As técnicas baseadas em gráfico visualizam regras de associação usando vértices e arestas onde os vértices são rotulados com nomes de itens, e conjuntos de itens ou regras são representados como um segundo conjunto de vértices. Itens estão conectados com item-sets/regras usando setas direcionadas. Setas apontando de itens para vértices de regra indicam itens LHS e uma seta de uma regra para um item indica o RHS. O tamanho e a cor dos vértices geralmente representam medidas de interesse.

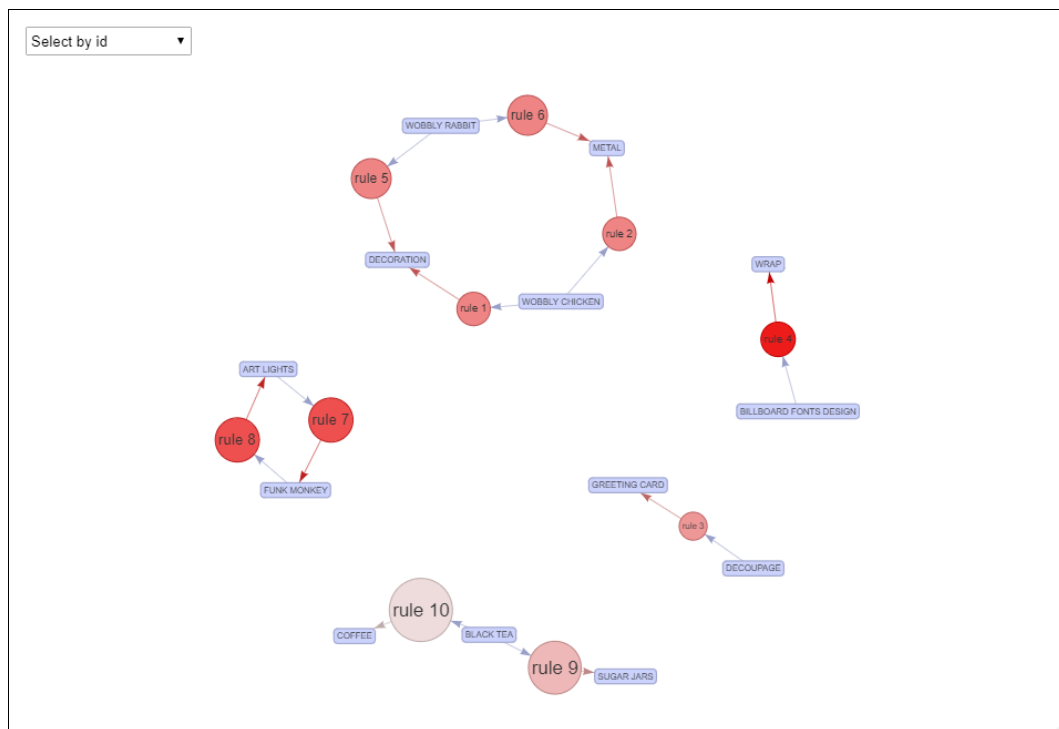
A geração de gráficos é uma ótima maneira de visualizar as regras, mas eles tendem a se tornar congestionados conforme o número de regras aumenta. Então é melhor visualizar um número menor de regras através dos gráficos.

Vamos selecionar as 10 regras de **subRules** que possuem a maior confiança e plotar um gráfico interativo.

```
top10subRules <- head(subRules, n = 10, by = "confidence")
```

Nota: Você pode fazer todas as suas plotagens interativas usando o parâmetro `engine=htmlwidget` no comando **plot**.

```
plot(top10subRules, method = "graph", engine = "htmlwidget")
```



Os gráficos **arulesViz** para conjuntos de regras de associação podem ser exportados no formato *GraphML* ou num arquivo *Graphviz* para ser explorado em ferramentas como **Gephi**. Por exemplo, as regras primeiras 1.000 regras com o maior Lift são exportadas por:

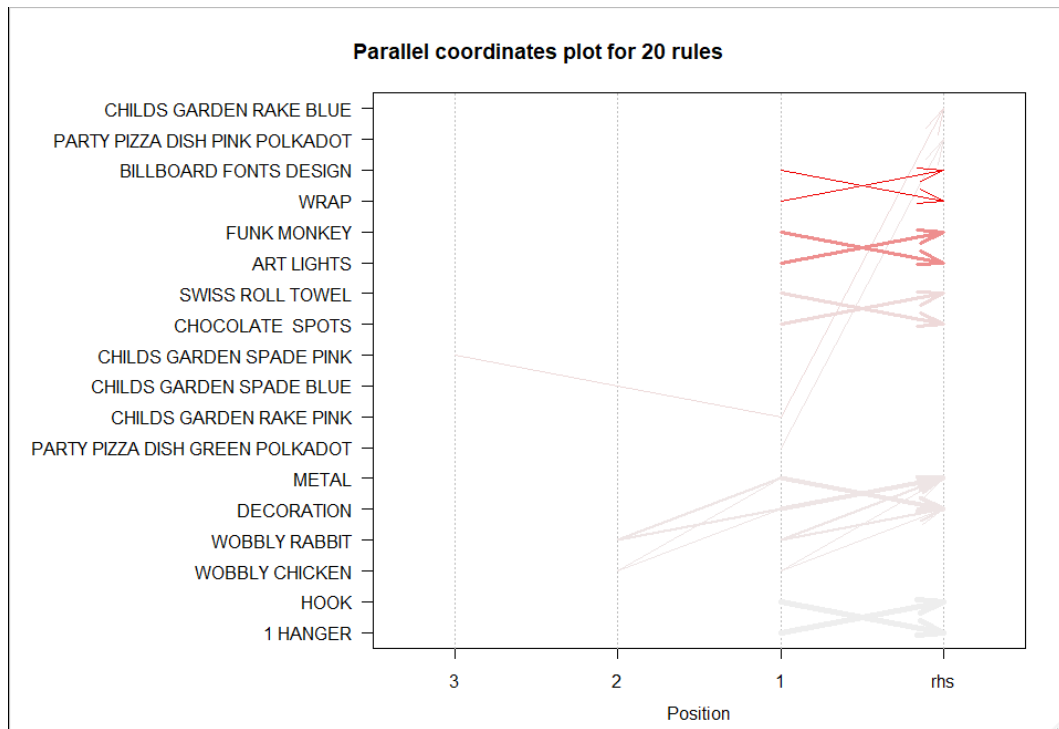
```
saveAsGraph(head(subRules, n = 1000, by = "lift"), file = "rules.graphml")
```

Representação de Regra Individual

Essa representação também é chamada de **Plotagem de coordenadas paralelas**. É útil visualizar que produtos junto com que itens causam que tipo das vendas.

Como mencionado acima, o RHS é o consequente ou o item que propomos que o cliente comprará; as posições estão no LHS onde 2 é a adição a mais recente a nossa cesta e 1 é o artigo que nós tivemos previamente.

```
# Filter top 20 rules with highest lift
subRules2<-head(subRules, n=20, by="lift")
plot(subRules2, method="paracoord")
```



Olhe para a seta de cima. Ela mostra que quando o cliente tem ' CHILDS GARDEN SPADE PINK ' e ' CHILDS GARDEN RAKE PINK ' no seu carrinho de compras, ele é susceptível a comprar também ' CHILDS GARDEN RAKE BLUE ' juntamente com estes produtos.

Etapa Final: Relatório de Elaboração do Laboratório

Você deve entregar um relatório com os resultados das etapas elaboradas neste laboratório no e-Disciplinas, para formatá-lo siga estas orientações:

1. Crie um documento Word e identifique-o com o nome do laboratório, data de elaboração e o seu nome ou da dupla que o elaborou;
2. Crie um tópico para cada resultado que você considerar relevante (manipulação de dados ou resultado de algum processamento) identificando-o com um título e uma breve explicação. Os resultados podem ser imagens de gráficos gerados ou de listas de valores ou dados de resultados obtidos. Não devem ser incluídos os scripts ou instruções de processamento utilizados, inclua apenas os resultados que você considerar relevantes.
3. No final do relatório crie um último tópico denominado "Conclusões" e elabore comentários, sugestões e conclusões sobre o que você pode aprender com a elaboração deste laboratório.

Conclusão

Parabéns! Você concluiu com sucesso o Laboratório de Regras de Associação; espero que tenha gostado e percebido o poder analítico que esta técnica oferece, especialmente para os analistas de Marketing.

Você aprendeu o algoritmo APRIORI, um dos algoritmos mais frequentemente usados na análise de dados. Você aprendeu quase tudo sobre a análise de regras de associação, seus recursos e suas aplicações no varejo como **Análise de Cesta de Compras**. Você agora deve ser capaz de implementar a análise de cesta de compras em R e apresentar regras de associação com algumas boas análises gráficas visuais para os seus clientes ou "chefes"!

Referências

- <https://datascienceplus.com/a-gentle-introduction-on-market-basket-analysis%E2%80%8A%E2%80%8Aassociation-rules/>
- https://en.wikipedia.org/wiki/Sparse_matrix
- <https://cran.r-project.org/web/packages/arulesViz/vignettes/arulesViz.pdf>
- <https://www.datacamp.com/Community/Tutorials/Market-Basket-Analysis-r>