



**Universidad Nacional Autónoma de México**  
Facultad de Ingeniería

# Reglas de asociación

**Guillermo Molero-Castillo**

guillermo.molero@ingenieria.unam.edu

Octubre, 2020

# **Sistemas de recomendación**

# Amazon

Cuando se pretende comprar un artículo, se muestra un apartado de artículos relacionados (recomendados). Lo que significa que se busca la compra de más cosas.

# Sistemas de recomendación

The screenshot shows the Amazon Mexico website interface. At the top, there's a navigation bar with the Amazon logo, a search bar, and links for account, orders, and Prime. Below this is a category bar with various product categories. The main content area features a product page for an ASUS VivoBook laptop. The product image is a black laptop with the ASUS logo on the screen. To the right of the image, there's a detailed description of the laptop, including its specifications (14" screen, Intel Core i3 10th Gen, 8GB RAM, 512GB SSD) and price (\$16,499.00). Below the description, there's a section for 'Acerca de este artículo' (About this item) with bullet points highlighting features like the Intel Optane memory, NumberPad, and the NanoEdge display. To the right of the product page, there's a sidebar with additional information, including the price, delivery options, and a section for 'Agregar un plan de protección' (Add a protection plan). At the bottom of the page, there's a section titled 'Productos relacionados con este artículo' (Products related to this item), which displays a row of recommended products, including various laptop bags and backpacks.

## Productos relacionados con este artículo

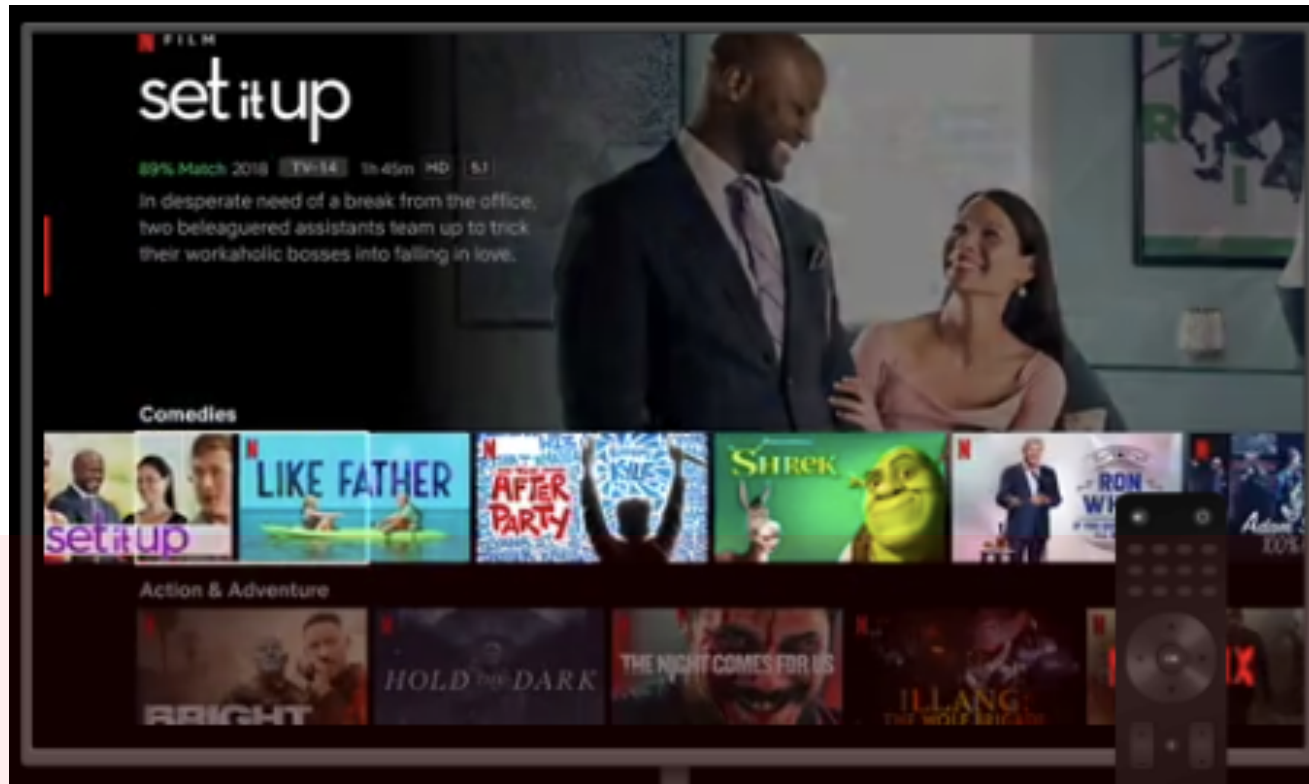
Patrocinado

Página 1 de 2



## Netflix

## Sistemas de recomendación

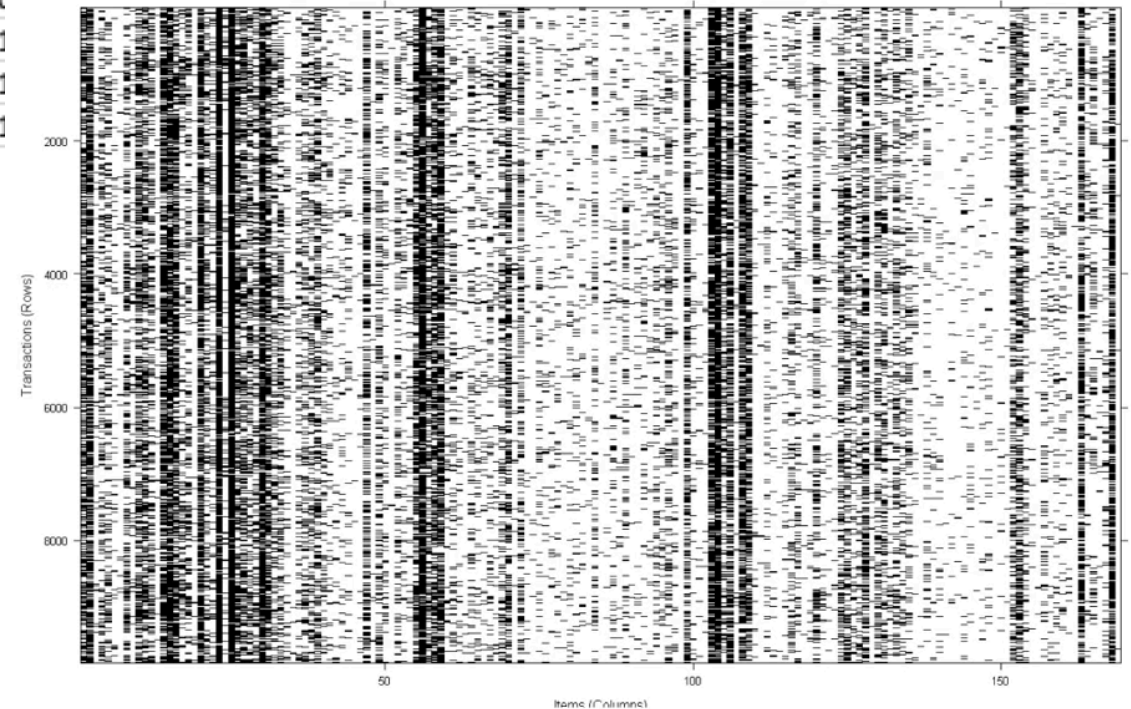


**Netflix** no busca una retribución directa como Amazon, lo que le interesa es **saber los gustos de los usuarios** para evitar el abandono de la suscripción mensual.

# Reglas de asociación

Reglas de asociación a partir de datos transaccionales

A	B	C	D	E	F	G	H	I	J	K
TID	Artículo 1	Artículo 2	Artículo 3	Artículo 4	Artículo 5	Artículo 6	Artículo 7	Artículo 8	Artículo 9	Artículo 10
1	1	0	1	1	0	0	0	0	0	0
2	1	1	0	0	0	0	-	-	-	-
3	0	0	0	0	1	1	-	-	-	-
4	1	1	0	0	0	1	-	-	-	-
5	1	0	0	0	0	1	-	-	-	-



# **Algoritmo Apriori**

# Reglas de asociación

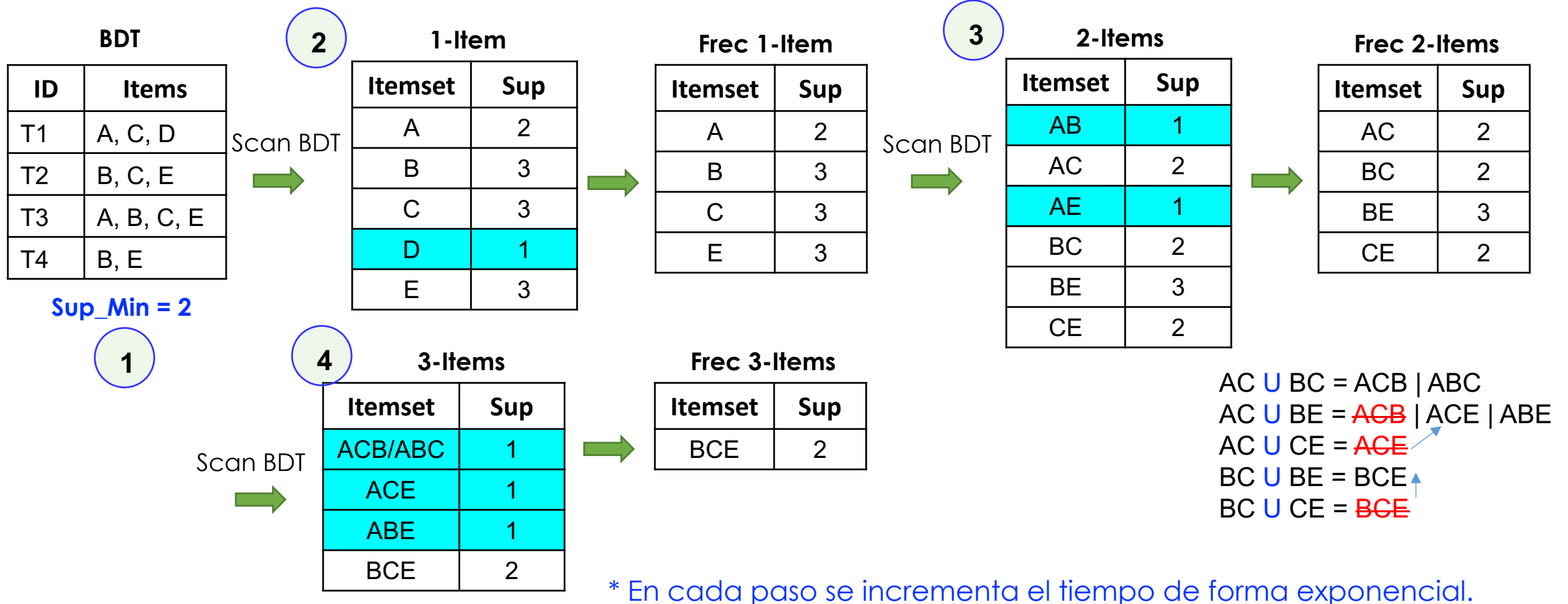
## Algoritmo Apriori

Para K productos

1. El usuario establece un criterio de soporte mínimo.
2. A continuación, se genera una lista de productos de **un ítem** que cumplan con el criterio de soporte mínimo.
3. Se utiliza la lista de un ítem para generar una lista de **dos ítems** que cumplan con el criterio de soporte mínimo.
4. Se utiliza la lista de dos ítems para generar una lista de **tres ítems**.
5. Se continua hasta construir un conjunto con el total de ítems disponibles (k).

# Reglas de asociación

## Algoritmo Apriori

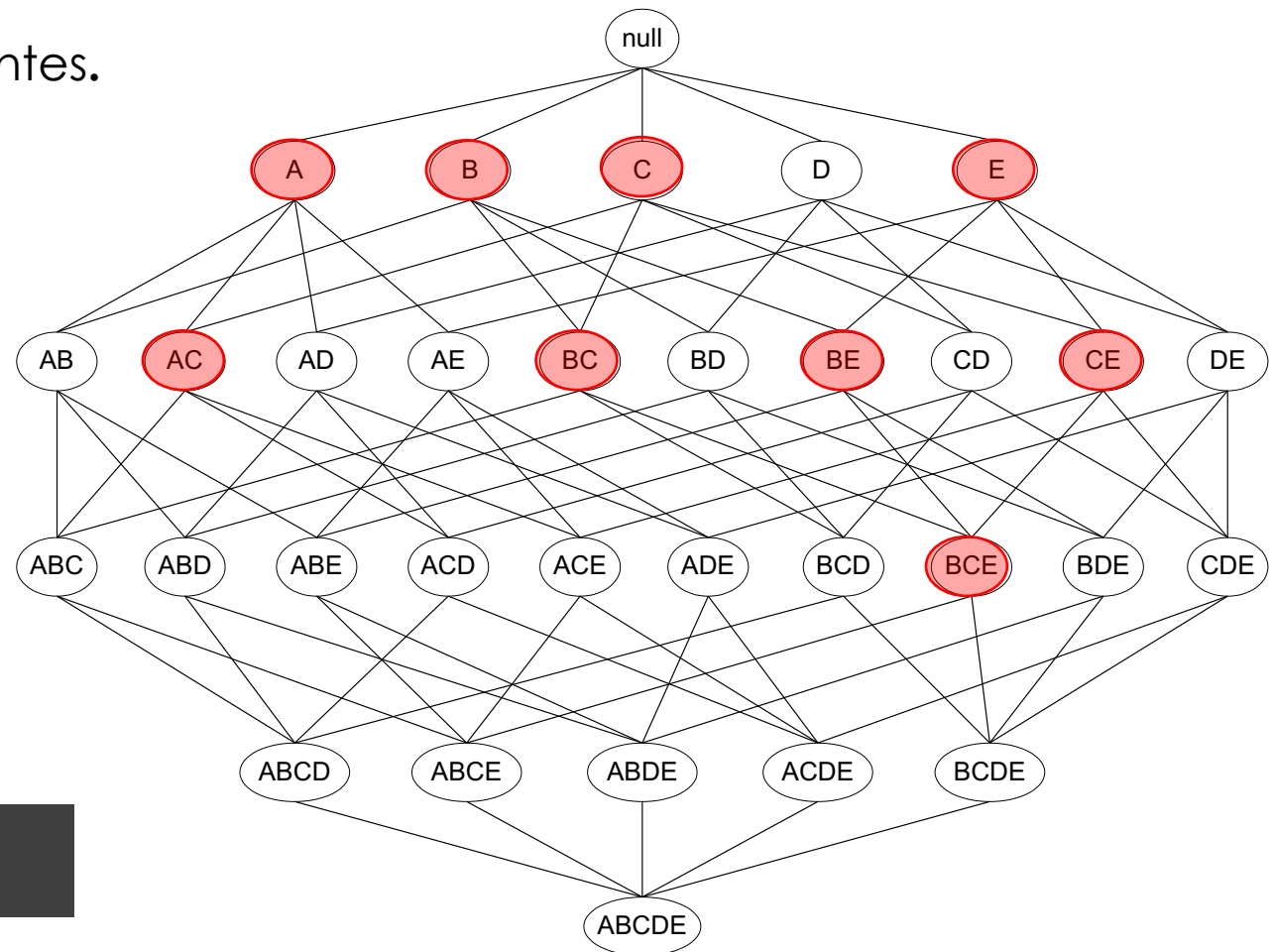




# Reglas de asociación

## Algoritmo Apriori

- Retomando la propiedad A priori.
- Se eliminan las reglas menos importantes.



Frecuencia Itemset:  
{A} {B} {C} {E} | {AC} {BC} {BE} {CE} | {BCE}

# Reglas de asociación

## Obtención de reglas significativas

La condición es encontrar las reglas que satisfagan cierto criterio, por ejemplo:

- Soporte mínimo (minsup)
- Confianza mínima (minconf)

### ► Soporte

$$\text{supp}(X \rightarrow Y) = \frac{\#(X \cup Y)}{n} = \text{supp}(X \cup Y)$$

### ► Confianza

$$\text{conf}(X \rightarrow Y) = \frac{\#(X \cup Y)}{\#X} = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)}$$

### ► Lift

$$\text{lift}(X \rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X) \text{supp}(Y)}$$

Rule:  $X \Rightarrow Y$

- Support =  $\frac{\text{Frequency}(X, Y)}{N}$
- Confidence =  $\frac{\text{Frequency}(X, Y)}{\text{Frequency}(X)}$
- Lift =  $\frac{\text{Support}}{\text{Support}(X) \times \text{Support}(Y)}$

# Práctica 1

# Reglas de asociación

## Caso 1

- Analizar las transacciones y obtener reglas significativas (patrones) de los productos vendidos en una tienda minorista.
  - Total 16 campos
  - Ítems (11 productos)
  - 1000 registros

ID	Valor	Pago	Sexo	Edad	Vegetales	CarneFresca	Leche	VerduraEnlatada	CarneEnlatada	ComidaCongelada	Cerveza	Vino	Refresco	Pescado	Dulces
10150	41.1354	TDebito	F	36	T	F	F	F	F	F	F	F	T	F	F
10236	42.0572	Efectivo	F	44	F	F	F	F	F	T	T	F	F	F	F
10360	27.4012	TCredito	M	20	F	F	F	T	F	T	T	F	F	T	F
10451	15.0174	TCredito	M	19	F	F	F	T	F	T	T	F	F	F	T
10609	14.2389	TCredito	F	41	T	F	F	F	F	F	F	F	F	T	F
10614	19.0425	Efectivo	M	25	F	F	F	F	F	F	F	F	T	F	F
10645	18.4803	TDebito	M	20	T	T	F	T	F	T	T	F	F	F	F
10717	47.2705	TDebito	M	23	T	T	F	F	F	F	F	F	F	T	F
10872	20.6176	Efectivo	M	36	F	F	F	T	F	T	T	F	F	T	F
10902	26.7182	TDebito	M	47	T	F	F	F	F	F	F	T	F	F	F
10915	12.4624	TDebito	F	22	T	F	T	F	T	T	F	F	F	T	F
10944	45.3278	TDebito	M	46	F	T	T	T	F	T	T	F	F	F	F
10987	37.7075	Efectivo	F	21	T	F	T	F	F	F	F	T	F	F	T
11119	29.4205	TCredito	F	27	F	F	F	F	F	F	F	F	T	T	T

# Reglas de asociación

## 1. Acceso a datos

```
Transacciones <- read.table("/Users/guille/Documents/1 FI-UNAM/1 Cursos/2021-1/1 IA2021-1/2  
CasosPracticos/1 RAsociacion/Transacciones.txt", header=T, sep="\t")
```

Transacciones

	ID	Valor	Pago	Sexo	Edad	Vegetales	CarneFresca	Leche	VerduraEnlatada	CarneEnlatada	ComidaCongelada	Cerveza	Vino	Refresco
1	10150	41.1354	TDebito	F	36	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
2	10236	42.0572	Efectivo	F	44	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
3	10360	27.4012	TCredito	M	20	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
4	10451	15.0174	TCredito	M	19	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
5	10609	14.2389	TCredito	F	41	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
6	10614	19.0425	Efectivo	M	25	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
7	10645	18.4803	TDebito	M	20	TRUE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
8	10717	47.2705	TDebito	M	23	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
9	10872	20.6176	Efectivo	M	36	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
10	10902	26.7182	TDebito	M	47	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
11	10915	12.4624	TDebito	F	22	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE
12	10944	45.3278	TDebito	M	46	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE
13	10987	37.7075	Efectivo	F	21	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
14	11119	29.4205	TCredito	F	27	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE
15	11220	23.6325	TDebito	M	27	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE

# Reglas de asociación

## 2. Exploración de la matriz de datos

summary(Transacciones)

ID	Valor	Pago	Sexo	Edad	Vegetales	CarneFresca	Leche	VerduraEnlatada
Min. : 10150	Min. :10.01	Efectivo:373	F:512	Min. :16.00	Mode :logical	Mode :logical	Mode :logical	Mode :logical
1st Qu.: 34577	1st Qu.:19.06	TCredito:207	M:488	1st Qu.:24.00	FALSE:701	FALSE:817	FALSE:823	FALSE:697
Median : 60630	Median :29.43	TDebito :420		Median :32.00	TRUE :299	TRUE :183	TRUE :177	TRUE :303
Mean : 60406	Mean :29.40			Mean :32.84	NA's :0	NA's :0	NA's :0	NA's :0
3rd Qu.: 87004	3rd Qu.:39.13			3rd Qu.:42.00				
Max. :109884	Max. :49.89			Max. :50.00				
CarneEnlatada	ComidaCongelada	Cerveza	Vino	Refresco	Pescado	Dulces		
Mode :logical	Mode :logical	Mode :logical	Mode :logical	Mode :logical	Mode :logical	Mode :logical		
FALSE:796	FALSE:698	FALSE:707	FALSE:713	FALSE:816	FALSE:708	FALSE:724		
TRUE :204	TRUE :302	TRUE :293	TRUE :287	TRUE :184	TRUE :292	TRUE :276		
NA's :0	NA's :0	NA's :0	NA's :0	NA's :0	NA's :0	NA's :0		

# Reglas de asociación

## 3. Cargar las librerías

```
install.packages("arules")    // En caso de no estar instalado
```

```
library(arulesViz)  
library(arules)
```

```
Reglas <- apriori(Transacciones[6:16], parameter= list(support=0.1, confidence = 0.5))
```

Parameter specification:

confidence	minval	smax	arem	aval	originalSupport	maxtime	support	minlen	maxlen	target	ext
0.5	0.1	1	none	FALSE	TRUE	5	0.1	1	10	rules	FALSE

Algorithmic control:

filter	tree	heap	memopt	load	sort	verbose
0.1	TRUE	TRUE	FALSE	TRUE	2	TRUE

Absolute minimum support count: 100

```
set item appearances ...[0 item(s)] done [0.00s].  
set transactions ...[11 item(s), 1000 transaction(s)] done [0.00s].  
sorting and recoding items ... [11 item(s)] done [0.00s].  
creating transaction tree ... done [0.00s].  
checking subsets of size 1 2 3 done [0.00s].  
writing ... [11 rule(s)] done [0.00s].  
creating S4 object ... done [0.00s].
```

# Reglas de asociación

## 4. Obtención de las reglas de asociación

Reglas

inspect(Reglas)

```
> Reglas
set of 11 rules
> inspect(Reglas)
```

	lhs	rhs	support	confidence	lift	count
[1]	{Dulces}	=> {Vino}	0.144	0.5217391	1.817906	144
[2]	{Vino}	=> {Dulces}	0.144	0.5017422	1.817906	144
[3]	{Cerveza}	=> {ComidaCongelada}	0.170	0.5802048	1.921208	170
[4]	{ComidaCongelada}	=> {Cerveza}	0.170	0.5629139	1.921208	170
[5]	{Cerveza}	=> {VerduraEnlatada}	0.167	0.5699659	1.881075	167
[6]	{VerduraEnlatada}	=> {Cerveza}	0.167	0.5511551	1.881075	167
[7]	{ComidaCongelada}	=> {VerduraEnlatada}	0.173	0.5728477	1.890586	173
[8]	{VerduraEnlatada}	=> {ComidaCongelada}	0.173	0.5709571	1.890586	173
[9]	{ComidaCongelada, Cerveza}	=> {VerduraEnlatada}	0.146	0.8588235	2.834401	146
[10]	{VerduraEnlatada, Cerveza}	=> {ComidaCongelada}	0.146	0.8742515	2.894873	146
[11]	{VerduraEnlatada, ComidaCongelada}	=> {Cerveza}	0.146	0.8439306	2.880309	146



# Reglas de asociación

## 4. Obtención de las reglas de asociación

```
OrdenarReglas <- sort(Reglas, by="confidence", decreasing=TRUE)
```

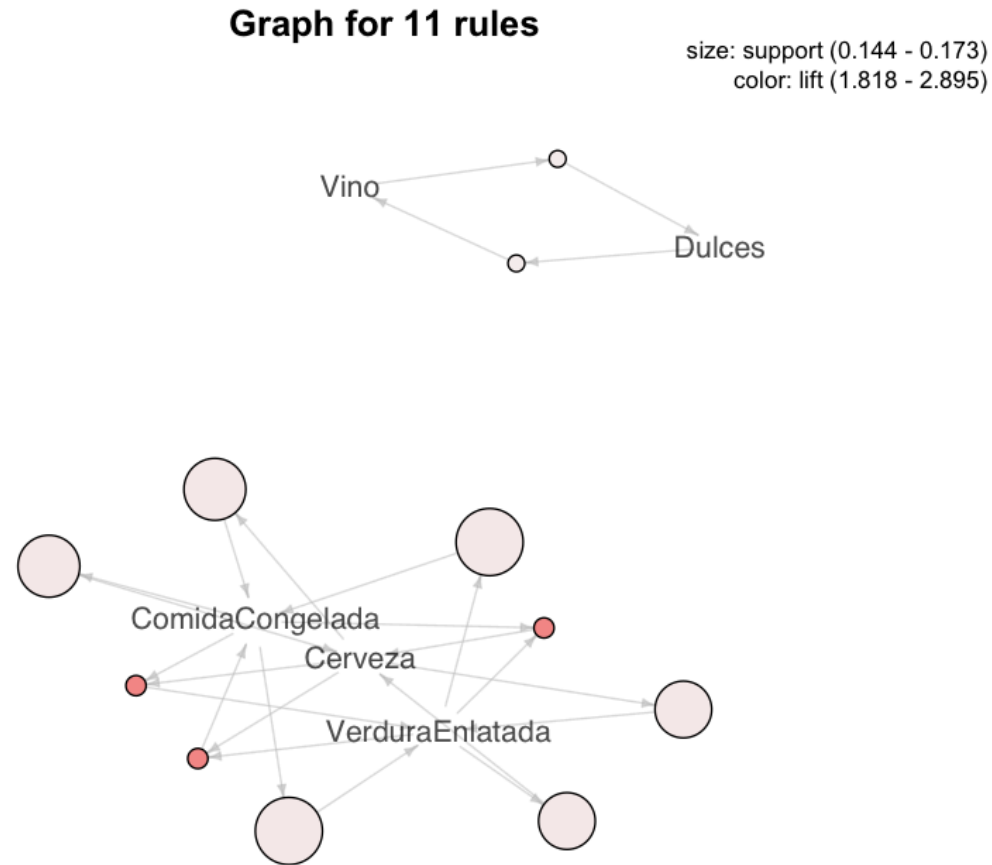
```
inspect(OrdenarReglas)
```

	lhs	rhs	support	confidence	lift	count
[1]	{VerduraEnlatada,Cerveza}	=> {ComidaCongelada}	0.146	0.8742515	2.894873	146
[2]	{ComidaCongelada,Cerveza}	=> {VerduraEnlatada}	0.146	0.8588235	2.834401	146
[3]	{VerduraEnlatada,ComidaCongelada}	=> {Cerveza}	0.146	0.8439306	2.880309	146
[4]	{Cerveza}	=> {ComidaCongelada}	0.170	0.5802048	1.921208	170
[5]	{ComidaCongelada}	=> {VerduraEnlatada}	0.173	0.5728477	1.890586	173
[6]	{VerduraEnlatada}	=> {ComidaCongelada}	0.173	0.5709571	1.890586	173
[7]	{Cerveza}	=> {VerduraEnlatada}	0.167	0.5699659	1.881075	167
[8]	{ComidaCongelada}	=> {Cerveza}	0.170	0.5629139	1.921208	170
[9]	{VerduraEnlatada}	=> {Cerveza}	0.167	0.5511551	1.881075	167
[10]	{Dulces}	=> {Vino}	0.144	0.5217391	1.817906	144
[11]	{Vino}	=> {Dulces}	0.144	0.5017422	1.817906	144

# Reglas de asociación

## 5. Gráficas de las reglas de asociación significativas

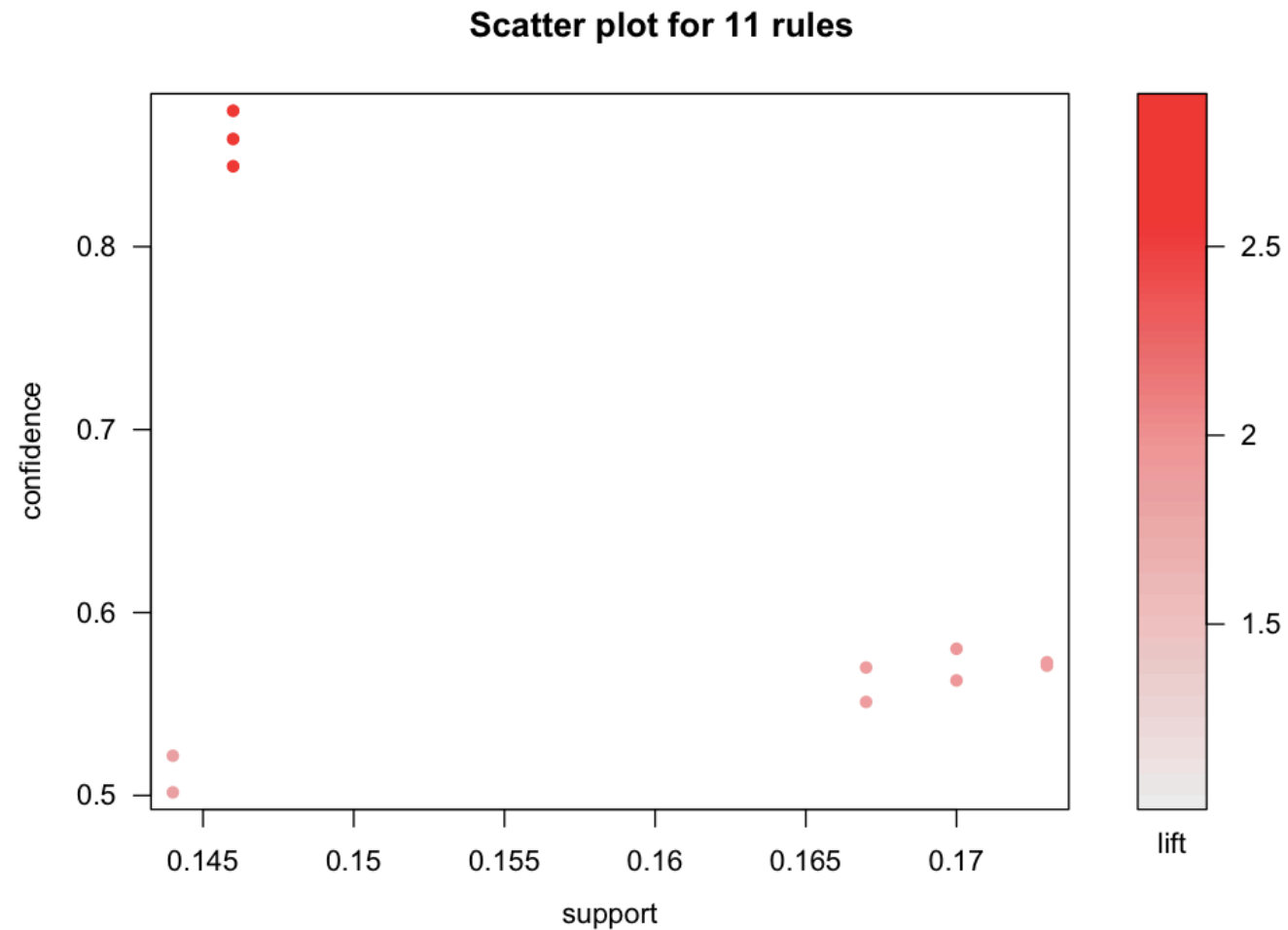
```
plot(OrdenarReglas, method="graph", control=list(type="items"))
```



# Reglas de asociación

## 5. Gráficas de las reglas de asociación significativas

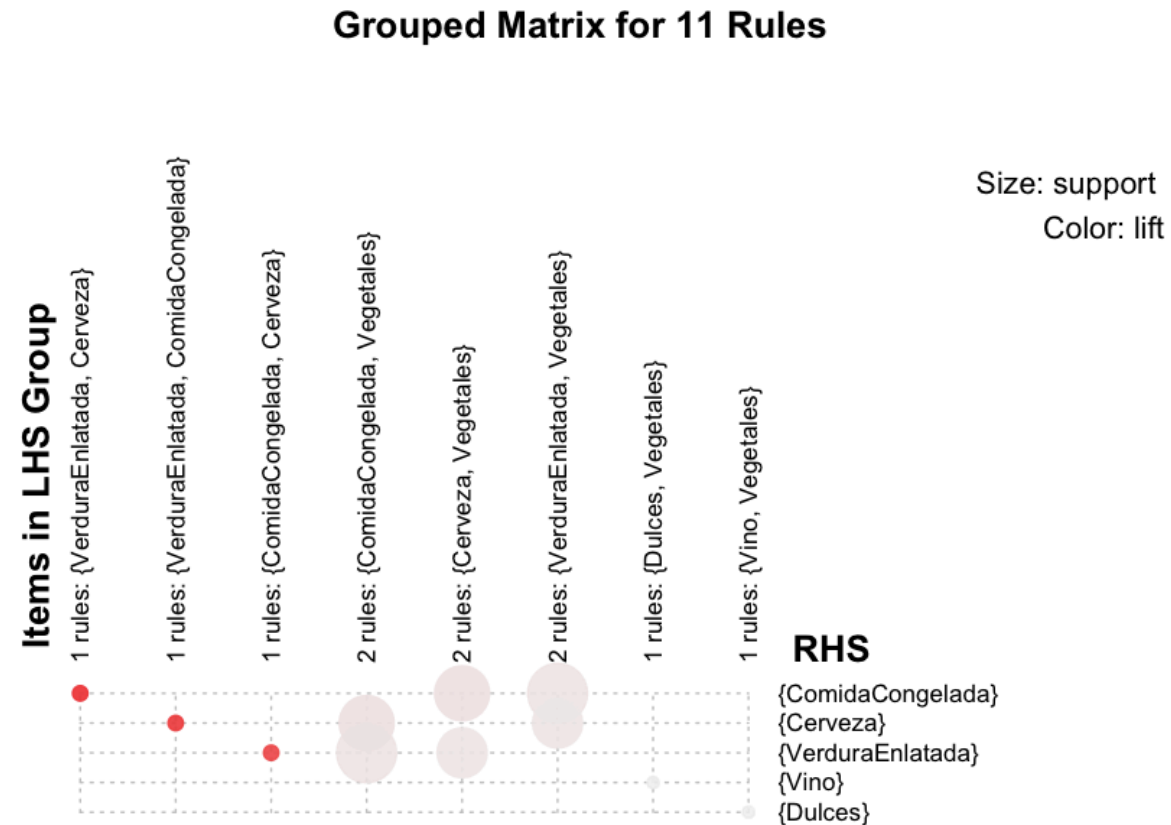
`plot(OrdenarReglas)`



# Reglas de asociación

## 5. Gráficas de las reglas de asociación significativas

```
plot(head(OrdenarReglas,11), method="grouped")
```



# Reglas de asociación

## 6. Interpretación de las reglas

Se muestran las reglas que cumplen con las restricciones definidas.  
La interpretación de las reglas fuertes y con la mayor confianza sería:

### Regla 1:

- Si los productos **VerduraEnlatada y Cerveza** son comprados, entonces hay un 87.4% de posibilidades de que también se compre **ComidaCongelada**.
- Esta regla se encuentra en 146 (14.6%) transacciones (soporte).
- La importancia (*Lift*) es 2.89, lo que significa que tener **VerduraEnlatada y Cerveza**, aumenta en un 2.89 veces la posibilidad de compra de **ComidaCongelada** (relación positiva).

	lhs	rhs	support	confidence	lift	count
[1]	{VerduraEnlatada,Cerveza}	=> {ComidaCongelada}	0.146	0.8742515	2.894873	146
[2]	{ComidaCongelada,Cerveza}	=> {VerduraEnlatada}	0.146	0.8588235	2.834401	146
[3]	{VerduraEnlatada,ComidaCongelada}	=> {Cerveza}	0.146	0.8439306	2.880309	146

# Reglas de asociación

## 7. Escribir las reglas de asociación

```
write(OrdenarReglas, file = "/Users/guille/Documents/1 FI-UNAM/1 Cursos/2021-1/1 IA2021-1/2 CasosPracticos/1  
RAsociacion/ReglasFuerres.csv", sep = ",", quote = TRUE, row.names = FALSE)
```

ReglasFuerres

rules	support	confidence	lift	count
{VerduraEnlatada,Cerveza} => {ComidaCongelada}	0.146	0.874251497005988	2.89487250664234	146
{ComidaCongelada,Cerveza} => {VerduraEnlatada}	0.146	0.858823529411765	2.83440108716754	146
{VerduraEnlatada,ComidaCongelada} => {Cerveza}	0.146	0.84393063583815	2.88030933733157	146
{Cerveza} => {ComidaCongelada}	0.17	0.580204778156997	1.92120787469204	170
{ComidaCongelada} => {VerduraEnlatada}	0.173	0.572847682119205	1.89058640963434	173
{VerduraEnlatada} => {ComidaCongelada}	0.173	0.570957095709571	1.89058640963434	173
{Cerveza} => {VerduraEnlatada}	0.167	0.569965870307167	1.88107547956161	167
{ComidaCongelada} => {Cerveza}	0.17	0.562913907284768	1.92120787469204	170
{VerduraEnlatada} => {Cerveza}	0.167	0.551155115511551	1.88107547956161	167
{Dulces} => {Vino}	0.144	0.521739130434783	1.81790637782154	144
{Vino} => {Dulces}	0.144	0.501742160278746	1.81790637782154	144

# Reglas de asociación

## En resumen

- Un **sistema de recomendación (SR)** es el filtrado de contenido (productos, servicios o eventos) con una determinada valoración.
- Este contenido representa información de mayor interés para los usuarios, ignorando toda aquella que no sea realmente útil.
- Por otro lado, un **SR** no es un sistema que predice las compras o consumo de los usuarios, sino una recomendación.
- Los **SR**, si no se implementan correctamente, pueden presentar problemas que afecten a la calidad de sus recomendaciones.
- Sobre todo a aquellos clientes con **gustos atípicos** y también debido a problemas de escasez de datos.

## **Práctica 2**



# Reglas de asociación

## Caso 2

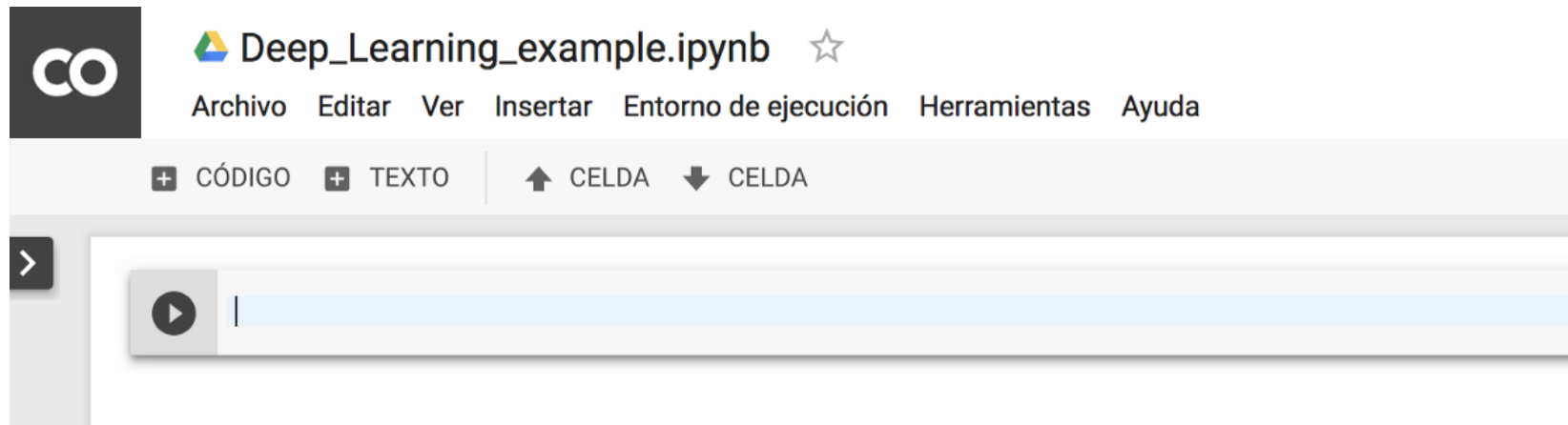
- Analizar las transacciones y obtener reglas significativas (patrones) de los productos vendidos en un comercio minorista en Francia.
  - Ítems (20 productos)
  - 7500 transacciones

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

**Fuente:** <https://drive.google.com/file/d/1y5DYn0dGoSbC22xowBq2d4po6h1JxcTQ/view?usp=sharing>

# Reglas de asociación

Google Colab



**Fuente:** <https://drive.google.com/file/d/1y5DYn0dGoSbC22xowBq2d4po6h1JxcTQ/view?usp=sharing>

# Reglas de asociación

## 1. Instalar y cargar funciones

### Reglas de asociación

Práctica

```
▶ !pip install apyori # Se instala el paquete Apyori

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from apyori import apriori
```

# Reglas de asociación

## 2. Importar el conjunto de datos

Se importa el conjunto de datos: store\_data.csv

Los datos son transacciones de un comercio de un periodo de una semana (7 días).

**Fuente:** iCollege | Kaggle <https://drive.google.com/file/d/1y5DYn0dGoSbC22xowBq2d4po6h1JxcTQ/view?usp=sharing>

```
[ ] from google.colab import files  
    files.upload()
```

Elegir archivos

No se eligió archivo

Upload widget is only available when the cell has been executed in the current browser session.

Please rerun this cell to enable.

Saving store\_data.csv to store\_data.csv

{'store\_data.csv': b'shrimp,almonds,avocado,vegetables mix,green grapes,whole weat flour,yams,cottage

```
[ ] Datos = pd.read_csv('store_data.csv')
```

# Reglas de asociación

## 3. Visualizar el conjunto de datos

Se utiliza la función head() para ver una muestra de los datos

```
[ ] Datos.head()
```

	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

# Reglas de asociación

## 3. Visualizar el conjunto de datos

[ ] Datos

	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt
0	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN
4	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...
7495	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7496	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN
7497	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7498	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7499	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN

7500 rows x 20 columns

# Reglas de asociación

## 3. Visualizar el conjunto de datos

### Observaciones:

- 1) No hay una fila de encabezado.
- 2) Se observa que el encabezado es la primera transacción.
- 3) Cada fila corresponde a una transacción y cada columna corresponde a un artículo comprado.
- 4) NaN indica que ese artículo no fue comprado en esa transacción.
- 5) Para evitar el problema del encabezado se agrega la opción `header=None` en la función `pd.read_csv`

```
[ ] Datos = pd.read_csv('store_data.csv', header=None)
    Datos.head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole wheat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

# Reglas de asociación

## 4. Procesamiento de datos

### Procesamiento de datos

La función Apriori de Python requiere que el conjunto de datos tenga la forma de una lista de listas, donde cada transacción es una *lista* interna dentro de una gran lista. Los datos actuales están en un dataframe de Pandas, por lo que, se requiere convertir en una lista de listas mediante el siguiente script:

```
[ ] registros = []  
    for i in range(0, 7501):  
        registros.append([str(Datos.values[i,j]) for j in range(0, 20)])
```

	0	1	2
0			
1			
2			
3			



## 5. Aplicación del algoritmo

### Algoritmo Apriori

La clase apriori de Python requiere algunos parámetros para funcionar: i) la lista de la que desea extraer reglas, ii) soporte mínimo (min\_support), iii) confianza mínima (min\_confidence), iv) elevación mínima (min\_lift), v) mínimo de elementos en la regla (min\_length).

**Consideraciones:** Obtener reglas para aquellos artículos que se compran al menos 50 veces al día, entonces,  $5 \times 7 = 35$  veces en una semana, entonces:

- i) El soporte mínimo se calcula de  $35/7500 = 0.0045$  (0.45%).
- ii) La confianza mínima para las reglas de 20%.
- iii) La elevación de 3.
- iv) El mínimo de elementos de 2 productos en las reglas.

**Nota.** En su mayoría, estos valores se eligen arbitrariamente, por lo que, se recomienda probar valores y analizar la diferencia en las reglas.

```
[ ] Reglas = apriori(registros, min_support=0.0045, min_confidence=0.2, min_lift=3, min_length=2)
```

# Reglas de asociación

## 6. Obtención de reglas

Se convierte las reglas encontradas por la clase apriori en una lista, puesto que es más fácil ver los resultados.

```
[ ] Resultados = list(Reglas)
```

## Ver los resultados

Primero, se obtiene el número total de reglas extraídas por la clase apriori.

```
[ ] print(len(Resultados))
```

48

# Reglas de asociación

## 6. Obtención de reglas

Son 48 reglas. A manera de ejemplo se imprime la primera regla:

```
[ ] print(Resultados[0])
```

```
RelationRecord(items=frozenset({'chicken', 'light cream'}), support=0.004532728969470737, ordered_statistics=
```

## 7. Imprimir las reglas

```
[ ] for item in Resultados:

    # Primer índice de la lista interna
    # Contiene un elemento base y agrega otro
    pair = item[0]
    items = [x for x in pair]
    print("Regla: " + items[0] + " -> " + items[1])

    # Segundo índice de la lista interna
    print("Soporte: " + str(item[1]))

    # Tercer índice de la lista interna
    print("Confianza: " + str(item[2][0][2]))
    print("Lift: " + str(item[2][0][3]))
    print("=====")
```

# Reglas de asociación

## 7. Imprimir las reglas

```
☞ Regla: chicken -> light cream
  Soporte: 0.004532728969470737
  Confianza: 0.29059829059829057
  Lift: 4.84395061728395
=====
Regla: escalope -> mushroom cream sauce
  Soporte: 0.005732568990801226
  Confianza: 0.3006993006993007
  Lift: 3.790832696715049
=====
Regla: escalope -> pasta
  Soporte: 0.005865884548726837
  Confianza: 0.3728813559322034
  Lift: 4.700811850163794
=====
Regla: ground beef -> herb & pepper
  Soporte: 0.015997866951073192
  Confianza: 0.3234501347708895
  Lift: 3.2919938411349285
```

La primera regla contiene dos elementos: **crema y pollo** que comúnmente se compran juntos.

Esto tiene sentido, las personas que compran **crema ligera** tienen cuidado con lo que comen, por lo que, es probable que compren **pollo**, en lugar de carne roja.

El soporte es de 0.0045, la confianza de 0.2905, la elevación de 4.84, esto es, **4.84** veces más probabilidades de que compren pollo.

## **Práctica extra-clase (Datos para el proyecto final)**

**Fuente en R:** <https://rpubs.com/Buczman/AssociationRules>

**Fuente en Python:** <https://www.kaggle.com/patelvishwa112/apriori-algorithm-on-grocery-market-data>

# Reglas de asociación

## Fuente de datos

Datos de compra recopilados en un mes de operación en una tienda de comestibles ().

- 9835 transacciones (327 transacciones por día)
- Aproximadamente 30 transacciones por hora en un día hábil de 12 horas.

<b>citrus fruit</b>	<b>semi-finished bread</b>	<b>margarine</b>	<b>ready soups</b>			
<b>tropical fruit</b>	yogurt	coffee				
<b>whole milk</b>						
<b>pip fruit</b>	yogurt	cream cheese	meat spreads			
<b>other vegetables</b>	whole milk	condensed milk	long life bakery product			
<b>whole milk</b>	butter	yogurt	rice	abrasive cleaner		
<b>rolls/buns</b>						
<b>other vegetables</b>	UHT-milk	rolls/buns	bottled beer	liquor (appetizer)		
<b>potted plants</b>						
<b>whole milk</b>	cereals					
<b>tropical fruit</b>	other vegetables	white bread	bottled water	chocolate		
<b>citrus fruit</b>	tropical fruit	whole milk	butter	curd	yogurt	flour

# Reglas de asociación

## Exploración de los datos

Exploración de la matriz de datos

`summary`(Consumibles)

```
transactions as itemMatrix in sparse format with
9835 rows (elements/itemsets/transactions) and
169 columns (items) and a density of 0.02609146
```

most frequent items:

whole milk	other vegetables	rolls/buns	soda	yogurt	(Other)
2513	1903	1809	1715	1372	34055

element (itemset/transaction) length distribution:  
sizes

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	26
2159	1643	1299	1005	855	645	545	438	350	246	182	117	78	77	55	46	29	14	14	9	11	4	6	1	1
27	28	29	32																					
1	1	3	1																					

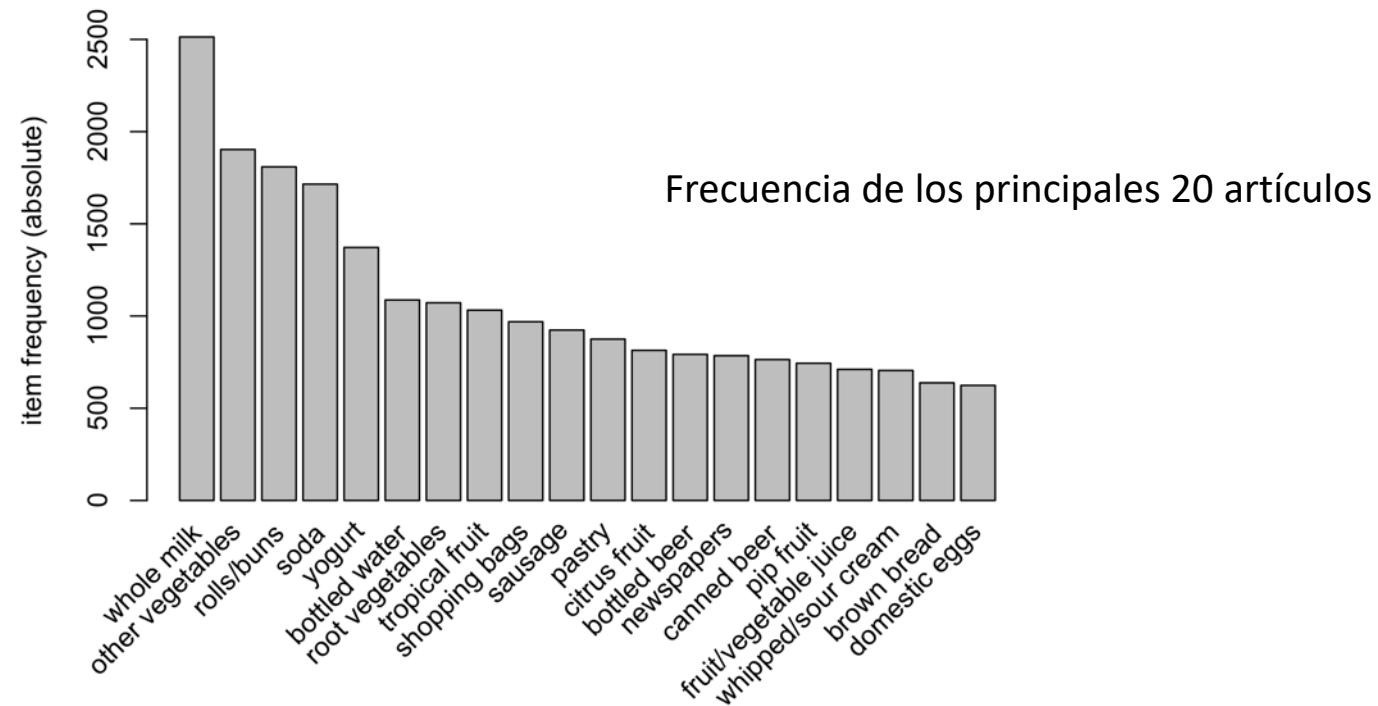


# Reglas de asociación

## Exploración de los datos

Explorar los datos antes de establecer las reglas:

```
itemFrequencyPlot(Consumibles, topN=20, type="absolute")
```



# Reglas de asociación

## Exploración de los datos

Visualización de la dispersión para las primeras 40 transacciones:

`image(Consumibles[1:40])`

