Proyecto Final (Association Rules)

Roberto Téllez Perezyera | Datamining | Profesor Marciano A. Moreno D.C.

5 de junio de 2021

Introducción

Este notebook contiene las actividades a realizar como parte del proyecto final de la materia Data Mining del Tecnológico de Monterrey, Campus Estado de México para el ciclo escolar enero/junio de 2021.

Las actividades incluyen ejecución, personalización y codificación de estatutos del lenguaje R con diversas liberarías de reglas de asociación, así como cálculos, análisis, investigación y exposición de conceptos relacionados.

La respuesta a todas y cada una de las actividades (implementación de código, notas y enlace a video) debe quedar registrada en el presente documento en formato R Markdown, exportada como documento PDF y publicada en el repositorio GitHub del alumno.

Fecha límite de entrega: sábado 5 de junio al cierre del día.

Recursos para la realización del proyecto

El proyecto podrá realizarse en un ambiente de desarrollo local o remoto, dependiendo de los recursos que disponga el alumno.

Recursos generales:

- Cuenta de usuario en GitHub.
- GitHub Desktop.
- Fork del repositorio principal de la clase
- Conectividad a Internet.
- Dispositivo de grabación de video.
- Cuenta para publicación de video (puede ser privado) como es el caso de YouTube.

Ambiente de desarrollo local:

- Procesaror Intel i5, 8 GB RAM, 50 GB HD.
- R Studio Desktop Open Source Edition.
- Sistema operativo soportado por R Studio.
- Proyecto de R Studio con el repositorio GitHub del alumno.

Ambiente de desarrollo remoto:

- Cuenta gratuita en R Studio Cloud.
- Espacio de trabajo creado a partir del repositorio GitHub del alumno.

Paquetes de R

- `git2r`
- `arulesViz` (incluye `arules`)
- `dplyr`

Conceptos

Chunk de código: región del documento identificada por triple tilde invertida.

```
#Chunk de código
```

Chunk de código sin personalización: El alumno deberá ejecutar el chunk de código sin mayor personalización.

```
#Chunk de código sin personalización.
print("Este es un chunk de código sin personalización.")
```

```
## [1] "Este es un chunk de código sin personalización."
```

Las actividades de codificación estarán identificadas con la clave **CODE-nn** previo al chunk y contarán con comentarios `#TODO:`.

CODE-01 Suma 2+3 Implementa la operación de adición de los enteros 2 y 3

```
#TODO: Implementa la operación de adición con los enteros: 2 y 3
2 + 3
```

```
## [1] 5
```

Notas en markdown: El alumno escribirá las notas requeridas en las secciones anotadas con la clave **NOTE-nn**

Escribe la fecha actual (**NOTE-01**): Fri Jun 04

Personalización del notebook

CODE-02 Datos generales Asigna tus datos a las siguientes variables:

```
NOMBRE_COMPLETO = "Roberto Tellez Perezyera"
DIGITOS_MATRICULA = 01374866
```

Instalación y carga de paquetes base

Instala los paquetes base de este notebook.

Los siguientes paquetes son requeridos para aspectos operativos de los ejercicios, no están relacionados con los temas de la materia. Si los paquetes no están instalados deberás retirar los comentarios de los siguientes estatutos y ejecutar el chunk. Coloca los comentarios nuevamente cuando hayas instalado los paquetes.

CODE-03 Instala `git2r`

```
#TODO: Retira los comentarios la primera vez para instalar los paquetes.
#install.packages("git2r")
# uncommented and installed
```

git2r es una interfaz para el sistema de control de versiones Git que usaremos para verificar que el número de commit del proyecto final sea el esperado. El valor esperado del commit será comunicado por separado.

```
library(git2r)
git2r::revparse_single(repository("."),"HEAD")
```

```
## [3b52c06] 2021-06-05: Complete project notebook and add dataset to local
folder
```

Instalación y carga de paquetes para los ejercicios

Instala los paquetes requeridos para este notebook.

En caso que los no tengas será necesario que retires los comentarios y ejecutes los comandos de la siguiente celda.

Tip: Coloca nuevamente en comentario las líneas de abajo en cuanto hayas instalado los paquetes.

CODE-04 Instala `arulesViz`

```
#TODO: Retira los comentarios la primera vez para instalar los paquetes.
#install.packages("arulesViz")
# uncommented and installed
```

Carga la librería arulesViz (la cual carga automáticamente arules).

```
library("arulesViz")
```

```
## Loading required package: arules
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'arules'
```

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

smallbasket: Fundamentos de Association Rules

Considera a `smallbasket` como la siguiente lista de transacciones, implementa el código necesario y responde a las solicitudes indicadas.

TID	items
Tr10	{beer, nuts, diapers}

Tr20 {beer, coffee, diapers}

Tr30 {beer, diapers, eggs}

Tr40 {nuts, eggs, milk}

Tr50 {nuts, coffee, diapers, eggs, milk}

Responde a las siguientes preguntas de smallbasket:

- Cantidad de transacciones (n) (**NOTE-02**): 5
- Frecuencia absoluta del item {beer} (**NOTE-03**): 3
- Frecuencia absoluta del item {nuts} (**NOTE-04**): 3
- Frecuencia absoluta del itemset {beer, nuts} (**NOTE-05**): 1
- Support del item {beer} (**NOTE-06**): 0.6, i.e., aparece en 3 de 5 transacciones
- Support del itemset {beer, nuts} (**NOTE-07**): 0.2, i.e., aparece en 1 de 5 transacciones

Implementaremos a `smallbasket` en R por medio de `list`.

```
smallbasket <- list(c("beer", "nuts", "diapers"),
                    c("beer", "coffee", "diapers"),
                    c("beer", "diapers", "eggs"),
                    c("nuts", "eggs", "milk"),
                    c("nuts", "coffee", "diapers", "eggs", "milk"))

names(smallbasket) <- paste("Tr", seq(from = 10, to = 50, by = 10), sep = "")
```

Crea un objeto `smalltx` de tipo `arules::transactions` a partir de la lista `smallbasket` el cual deberá tener la siguiente estructura lógica:

TID	beer	nuts	diapers	coffee	eggs	milk
Tr10	1	1	1	0	0	0
Tr20	1	0	1	1	0	0
Tr30	1	0	1	0	1	0
Tr40	0	1	0	0	1	1
Tr50	0	1	1	1	1	1

CODE-05 Crea una variable `smalltx` que sea un objeto `transactions` a partir de los datos en `smallbasket`.

```
#TODO: Modifica el código de abajo para transformar smallbaskets a un objeto
de tipo transactions
smalltx <- as(smallbasket, "transactions")
```

Verifica que smalltx sea de tipo transactions.

```
#class(smalltx)[1]
class(smalltx)[1]=="transactions"
```

```
## [1] TRUE
```

Implementa en el chunk de abajo el código para visualizar la información general del objeto `smalltx` por medio de la función `summary()`, visualiza la salida **R Console** y responde en línea en el markdown a las preguntas de abajo.

```
summary(smalltx)
```

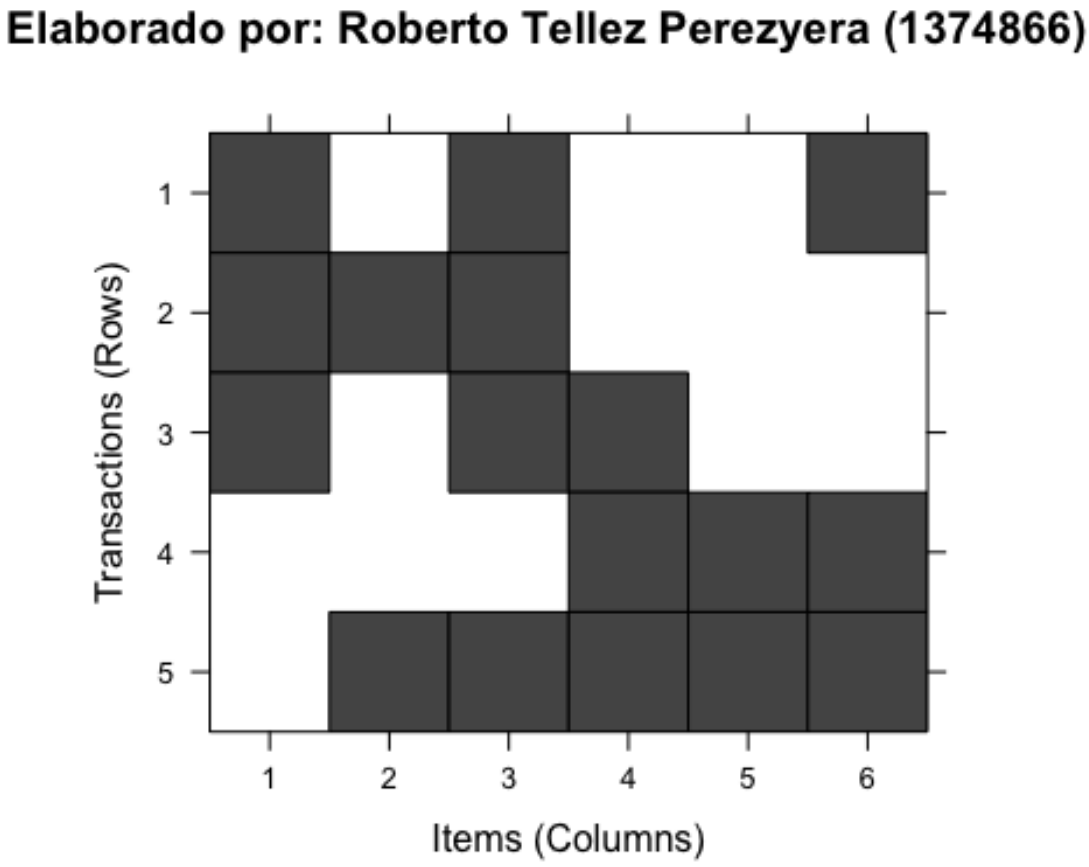
```
## transactions as itemMatrix in sparse format with
##  5 rows (elements/itemsets/transactions) and
##  6 columns (items) and a density of 0.5666667
##
## most frequent items:
## diapers    beer    eggs    nuts  coffee (Other)
##       4       3       3       3       2       2
##
## element (itemset/transaction) length distribution:
## sizes
## 3 5
## 4 1
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     3.0     3.0     3.0     3.4     3.0     5.0
##
## includes extended item information - examples:
##    labels
## 1    beer
## 2  coffee
## 3 diapers
##
## includes extended transaction information - examples:
##   transactionID
## 1          Tr10
## 2          Tr20
## 3          Tr30
```

- Número de renglones (**NOTE-08**): 5 rows
- Número de columnas (**NOTE-09**): 6 cols
- Densidad de la matriz (**NOTE-10**): 0.5666667
- Media aritmética del número de elementos (itemsets) por transacción (**NOTE-11**): 3.4

Visualiza la matriz de items y transacciones.

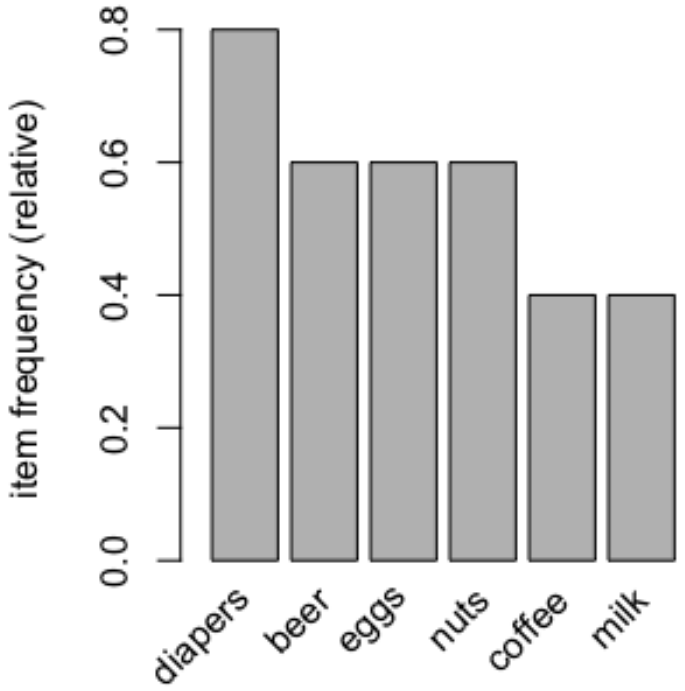
```
try(
  plot(smalltx, main = paste0("Elaborado por: ", NOMBRE_COMPLETO, " (",
DIGITOS_MATRICULA, ")" ))
)
```

```
## Warning in plot.itemMatrix(smalltx, main = paste0("Elaborado por: ",
## NOMBRE_COMPLETO, : Use image() instead of plot().
```



Visualiza el gráfico de frecuencia de items.

```
try(
  itemFrequencyPlot(smalltx, topN=10, main = paste0("Elaborado por: ",
NOMBRE_COMPLETO, " (", DIGITOS_MATRICULA, ")" ))
)
```



Invoca el algoritmo **apriori** de `arules` con las transacciones de `smalltx` y asignando el resultado a `smallrules`.

```
smallrules <- apriori(smalltx)
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    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: 0
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[6 item(s), 5 transaction(s)] done [0.00s].
## sorting and recoding items ... [6 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
```

```
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [46 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].

#parameter = list(support = 0.01, confidence = 0.01)
```

Menciona los valores por omisión de algoritmo apriori de la librería arules:

- support (mínimo) (**NOTE-12**): 0.1
- confidence (mínimo) (**NOTE-13**): 0.8
- items (máximo) (**NOTE-14**): 10
- tiempo de verificación de subsets (máximo) (**NOTE-15**): 5 sec

Invoca `summary()` con `smallrules`, consulta los resultados en **R Console** y responde a las preguntas de abajo.

Tip: Para mayor legibilidad emplea el comando **Show in new window**

```
summary(smallrules)

## set of 46 rules
##
## rule length distribution (lhs + rhs):sizes
##  1  2  3  4  5
##  1  4 18 18  5
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.500   3.478   4.000   5.000
##
## summary of quality measures:
##     support         confidence        coverage           lift
##  Min.   :0.2000   Min.   :0.8000   Min.   :0.2000   Min.   :1.000
##  1st Qu.:0.2000   1st Qu.:1.0000   1st Qu.:0.2000   1st Qu.:1.250
##  Median :0.2000   Median :1.0000   Median :0.2000   Median :1.667
##  Mean   :0.2478   Mean   :0.9957   Mean   :0.2522   Mean   :1.779
##  3rd Qu.:0.2000   3rd Qu.:1.0000   3rd Qu.:0.2000   3rd Qu.:2.500
##  Max.   :0.8000   Max.   :1.0000   Max.   :1.0000   Max.   :2.500
##      count
##  Min.   :1.000
##  1st Qu.:1.000
##  Median :1.000
##  Mean   :1.239
##  3rd Qu.:1.000
##  Max.   :4.000
##
## mining info:
##     data ntransactions support confidence
##  smalltx             5     0.1        0.8
```

- Cantidad de reglas producidas (**NOTE-16**): 46 reglas

- Media aritmética de support (**NOTE-17**): 0.2478
- Media aritmética de confidence (**NOTE-18**): 0.9957

Inspecciona `smallrules` y responde a las preguntas a continuación.

```
inspect(smallrules)
```

```
##      lhs                          rhs       support confidence coverage
## [1]  {}                        => {diapers} 0.8     0.8        1.0
## [2]  {coffee}                  => {diapers} 0.4     1.0        0.4
## [3]  {milk}                    => {nuts}    0.4     1.0        0.4
## [4]  {milk}                    => {eggs}    0.4     1.0        0.4
## [5]  {beer}                    => {diapers} 0.6     1.0        0.6
## [6]  {coffee,milk}             => {nuts}    0.2     1.0        0.2
## [7]  {coffee,nuts}             => {milk}    0.2     1.0        0.2
## [8]  {coffee,milk}             => {eggs}    0.2     1.0        0.2
## [9]  {coffee,eggs}             => {milk}    0.2     1.0        0.2
## [10] {coffee,milk}             => {diapers} 0.2     1.0        0.2
## [11] {diapers,milk}            => {coffee}  0.2     1.0        0.2
## [12] {beer,coffee}             => {diapers} 0.2     1.0        0.2
## [13] {coffee,nuts}             => {eggs}    0.2     1.0        0.2
## [14] {coffee,eggs}             => {nuts}    0.2     1.0        0.2
## [15] {coffee,nuts}             => {diapers} 0.2     1.0        0.2
## [16] {coffee,eggs}             => {diapers} 0.2     1.0        0.2
## [17] {milk,nuts}               => {eggs}    0.4     1.0        0.4
## [18] {eggs,milk}               => {nuts}    0.4     1.0        0.4
## [19] {eggs,nuts}               => {milk}    0.4     1.0        0.4
## [20] {diapers,milk}            => {nuts}    0.2     1.0        0.2
## [21] {diapers,milk}            => {eggs}    0.2     1.0        0.2
## [22] {beer,nuts}               => {diapers} 0.2     1.0        0.2
## [23] {beer,eggs}               => {diapers} 0.2     1.0        0.2
## [24] {coffee,milk,nuts}        => {eggs}    0.2     1.0        0.2
## [25] {coffee,eggs,milk}        => {nuts}    0.2     1.0        0.2
## [26] {coffee,eggs,nuts}        => {milk}    0.2     1.0        0.2
## [27] {coffee,milk,nuts}        => {diapers} 0.2     1.0        0.2
## [28] {coffee,diapers,milk}     => {nuts}    0.2     1.0        0.2
## [29] {coffee,diapers,nuts}     => {milk}    0.2     1.0        0.2
## [30] {diapers,milk,nuts}       => {coffee}  0.2     1.0        0.2
## [31] {coffee,eggs,milk}        => {diapers} 0.2     1.0        0.2
## [32] {coffee,diapers,milk}     => {eggs}    0.2     1.0        0.2
## [33] {coffee,diapers,eggs}     => {milk}    0.2     1.0        0.2
## [34] {diapers,eggs,milk}       => {coffee}  0.2     1.0        0.2
## [35] {coffee,eggs,nuts}        => {diapers} 0.2     1.0        0.2
## [36] {coffee,diapers,nuts}     => {eggs}    0.2     1.0        0.2
## [37] {coffee,diapers,eggs}     => {nuts}    0.2     1.0        0.2
## [38] {diapers,eggs,nuts}       => {coffee}  0.2     1.0        0.2
## [39] {diapers,milk,nuts}       => {eggs}    0.2     1.0        0.2
## [40] {diapers,eggs,milk}       => {nuts}    0.2     1.0        0.2
## [41] {diapers,eggs,nuts}       => {milk}    0.2     1.0        0.2
## [42] {coffee,eggs,milk,nuts}   => {diapers} 0.2     1.0        0.2
```

```
## [43] {coffee,diapers,milk,nuts} => {eggs}    0.2     1.0        0.2
## [44] {coffee,diapers,eggs,milk} => {nuts}    0.2     1.0        0.2
## [45] {coffee,diapers,eggs,nuts} => {milk}    0.2     1.0        0.2
## [46] {diapers,eggs,milk,nuts}   => {coffee}  0.2     1.0        0.2
##      lift     count
## [1]  1.000000 4
## [2]  1.250000 2
## [3]  1.666667 2
## [4]  1.666667 2
## [5]  1.250000 3
## [6]  1.666667 1
## [7]  2.500000 1
## [8]  1.666667 1
## [9]  2.500000 1
## [10] 1.250000 1
## [11] 2.500000 1
## [12] 1.250000 1
## [13] 1.666667 1
## [14] 1.666667 1
## [15] 1.250000 1
## [16] 1.250000 1
## [17] 1.666667 2
## [18] 1.666667 2
## [19] 2.500000 2
## [20] 1.666667 1
## [21] 1.666667 1
## [22] 1.250000 1
## [23] 1.250000 1
## [24] 1.666667 1
## [25] 1.666667 1
## [26] 2.500000 1
## [27] 1.250000 1
## [28] 1.666667 1
## [29] 2.500000 1
## [30] 2.500000 1
## [31] 1.250000 1
## [32] 1.666667 1
## [33] 2.500000 1
## [34] 2.500000 1
## [35] 1.250000 1
## [36] 1.666667 1
## [37] 1.666667 1
## [38] 2.500000 1
## [39] 1.666667 1
## [40] 1.666667 1
## [41] 2.500000 1
## [42] 1.250000 1
## [43] 1.666667 1
## [44] 1.666667 1
```

```
## [45] 2.500000 1
## [46] 2.500000 1
```

- ¿Cuál es la interpretación de la regla `{} => {diapers}` en la que se aprecia la ausencia del LHS? (**NOTE-19**): por cómo se puede leer la regla en 'español llano', diríamos que a partir de no comprar items, se comprarán pañales. No obstante, dado el lift de exactamente 1, estamos seguros de que estos LHS y RHS son independientes, i.e., NO constituyen una regla.
- Calcula el valor de la métrica support para la regla `{beer} => {nuts}` (**NOTE-20**): 0.2
- Calcula el valor de la métrica confidence para la regla `{beer} => {nuts}` (**NOTE-21**): 0.333
- ¿Por qué no aparece listada la regla `{beer} => {nuts}`? (**NOTE-22**): Notemos que se corrió el algoritmo apriori con todos sus parámetros por defecto, i.e., no alteramos ninguno. Quedó documentado en NOTE-12 y NOTE-13 que los valores de soporte y confidence mínimos que deben tener las reglas a generar son de 0.1 (para supp) y 0.8 (para conf). Si bien el valor de support de `{beer} => {nuts}` es aceptable, su valor de confidence está por debajo del threshold de mínimo 0.8 definido por defecto, por lo cual no se considera en el listado de reglas.

MSSD: El dataset de sesiones de streaming

The Music Streaming Sessions Dataset fue desarrollado por Spotify para promover la investigación en modelado de escucha por parte de usuarios, interacciones en streaming, recuperación de información musical (MIR, por sus siglas en inglés) y recomendaciones con base a sesiones secuenciales.

Analiza una extracción del dataset MSSD por medio de reglas de asociación.

Instala el paquete dplyr para llevar preparar los datos de MSSD de tal forma que puedan ser analizados por medio de `arules`.

CODE-06 Instala el paquete dplyr

```
#TODO: Retira los comentarios la primera vez para instalar los paquetes.
#install.packages("dplyr")
# uncommented and installed
```

Carga la librería dplyr.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

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

```
## The following object is masked from 'package:git2r':
##
##     pull

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

Lee el dataset MSSD y verifica su estructura.

```
#mssddf <- read.csv(url("https://storage.googleapis.com/data-mining-
202104/mssd-log_mini.csv"))
#../../../data/mssd-log_mini.csv

# save to same directory and read from there
mssddf <- read.csv('./mssd-log_mini.csv')
str(mssddf)

## 'data.frame':    167880 obs. of  21 variables:
##  $ session_id                    : chr  "0_00006f66-33e5-4de7-a324-
2d18e439fc1e" "0_00006f66-33e5-4de7-a324-2d18e439fc1e" "0_00006f66-33e5-4de7-
a324-2d18e439fc1e" "0_00006f66-33e5-4de7-a324-2d18e439fc1e" ...
##  $ session_position              : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ session_length                : int  20 20 20 20 20 20 20 20 20 20 ...
##  $ track_id_clean                : chr  "t_0479f24c-27d2-46d6-a00c-
7ec928f2b539" "t_9099cd7b-c238-47b7-9381-f23f2c1d1043" "t_fc5df5ba-5396-49a7-
8b29-35d0d28249e0" "t_23cff8d6-d874-4b20-83dc-94e450e8aa20" ...
##  $ skip_1                        : chr  "false" "false" "false" "false"
...
##  $ skip_2                        : chr  "false" "false" "false" "false"
...
##  $ skip_3                        : chr  "false" "false" "false" "false"
...
##  $ not_skipped                   : chr  "true" "true" "true" "true" ...
##  $ context_switch                : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ no_pause_before_play          : int  0 1 1 1 1 1 1 1 1 1 ...
##  $ short_pause_before_play       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ long_pause_before_play        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hist_user_behavior_n_seekfwd  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hist_user_behavior_n_seekback : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ hist_user_behavior_is_shuffle : chr  "true" "true" "true" "true" ...
##  $ hour_of_day                   : int  16 16 16 16 16 16 16 16 16 16 ...
##  $ date                          : chr  "2018-07-15" "2018-07-15" "2018-
07-15" "2018-07-15" ...
##  $ premium                       : chr  "true" "true" "true" "true" ...
##  $ context_type                  : chr  "editorial_playlist"
"editorial_playlist" "editorial_playlist" "editorial_playlist" "editorial_playlist" ...
```

```
##  $ hist_user_behavior_reason_start: chr  "trackdone" "trackdone"
"trackdone" "trackdone" ...
##  $ hist_user_behavior_reason_end  : chr  "trackdone" "trackdone"
"trackdone" "trackdone" ...
```

Inspecciona la columna `session_id`.

```
summary(mssddf$session_id)

##    Length     Class      Mode
##    167880 character character

head(mssddf$session_id)

## [1] "0_00006f66-33e5-4de7-a324-2d18e439fc1e"
## [2] "0_00006f66-33e5-4de7-a324-2d18e439fc1e"
## [3] "0_00006f66-33e5-4de7-a324-2d18e439fc1e"
## [4] "0_00006f66-33e5-4de7-a324-2d18e439fc1e"
## [5] "0_00006f66-33e5-4de7-a324-2d18e439fc1e"
## [6] "0_00006f66-33e5-4de7-a324-2d18e439fc1e"

length(unique(mssddf$session_id))

## [1] 10000
```

Inspecciona la columna `track_id_clean`.

```
summary(mssddf$track_id_clean)

##    Length     Class      Mode
##    167880 character character

head(mssddf$track_id_clean)

## [1] "t_0479f24c-27d2-46d6-a00c-7ec928f2b539"
## [2] "t_9099cd7b-c238-47b7-9381-f23f2c1d1043"
## [3] "t_fc5df5ba-5396-49a7-8b29-35d0d28249e0"
## [4] "t_23cff8d6-d874-4b20-83dc-94e450e8aa20"
## [5] "t_64f3743c-f624-46bb-a579-0f3f9a07a123"
## [6] "t_c815228b-3212-4f9e-9d4f-9cb19b248184"

length(unique(mssddf$track_id_clean))

## [1] 50704
```

Consulta el paper The Music Streaming Sessions Dataset y responde a las siguientes preguntas:

- ¿Cuál es el propósito de `session id`? (**NOTE-23**): Al ser el "identificador único de sesión", funciona como la llave primaria (un identificador único por definición) para cada registro en el dataset.
- ¿Cuál es el propósito de `track id`? (**NOTE-24**): Es el identificador único de cada pista. Podemos asumir que se espera ver registros repetidos, i.e., tiene sentido que

uno o más usuarios en la misma o varias sesiones escuchen la misma canción (track de audio). En especial si es popular.

- ¿Consideras que las otras columnas del dataset son aplicables para análisis por medio de reglas de asociación, justifica tu respuesta? (**NOTE-25**): Quizá no para análisis hechos estrictamente con reglas de asociación. E.g., se reporta que en el dataset también se ofrecen características del audio y metadatos, pero estos no tienen gran utilidad en el contexto de una metodología que explora cómo la presencia de ciertos items en un set (LHS) conlleva -y qué tan probablemente o con qué tanta confianza lo hace- a que otros items específicos (contenidos en el RHS de una regla) se adicionen al set. Por otra parte, se podría hacer más específico el scope del análisis considerando generar reglas solo para usuarios que son premium, la fecha (pensando no solo en días puntuales sino en temporadas) y la hora del día. Todos estos campos son provistos en columnas del dataset.

Inspecciona la estructura de `y`, el dataset preparado.

```
# consider tracks grouped by session_id, i.e., you get the track id's per
individual listening session
x <- mssddf %>% select(session_id, track_id_clean) %>% group_by(session_id)
y <- as.data.frame(x)
str(y)
```

```
## 'data.frame':    167880 obs. of  2 variables:
##  $ session_id    : chr  "0_00006f66-33e5-4de7-a324-2d18e439fc1e"
"0_00006f66-33e5-4de7-a324-2d18e439fc1e" "0_00006f66-33e5-4de7-a324-
2d18e439fc1e" "0_00006f66-33e5-4de7-a324-2d18e439fc1e" ...
##  $ track_id_clean: chr  "t_0479f24c-27d2-46d6-a00c-7ec928f2b539"
"t_9099cd7b-c238-47b7-9381-f23f2c1d1043" "t_fc5df5ba-5396-49a7-8b29-
35d0d28249e0" "t_23cff8d6-d874-4b20-83dc-94e450e8aa20" ...
```

Descarga la librería `dplyr`

```
#Descargamos dplyr porque enmascara otras funciones y no es requerida en lo
sucesivo
detach(package:dplyr)
```

Genera la representación de transacciones para `arules` del dataset MSSD y visualiza la salida **R Console**.

```
mssdtx <- as(split(y[,"track_id_clean"], y["session_id"]), "transactions")
```

```
## Warning in asMethod(object): removing duplicated items in transactions
```

```
#Trabajaremos con una muestra del 80% de las transacciones
numSessions <- round(nrow(mssdtx) * 0.8)
set.seed(DIGITOS_MATRICULA)
mssdtx <- sample(mssdtx, numSessions)
summary(mssdtx)
```

```
## transactions as itemMatrix in sparse format with
##  8000 rows (elements/itemsets/transactions) and
```

```
##  50704 columns (items) and a density of 0.0002895013
##
## most frequent items:
## t_bacf06d3-9185-4183-84ea-ff0db51475ce t_5718ab08-3a15-4d3f-9e63-
42b2f6805e31
##                                    883
593
## t_8c4d29b1-e0bf-464c-88f7-ac19240cbba0 t_77b02acb-1b1f-4b36-b8fc-
2c3e01892b9a
##                                    537
500
## t_a66ea088-b357-449a-8a1e-64dd0b8d6cb5
(Other)
##                                    496
114422
##
## element (itemset/transaction) length distribution:
## sizes
##    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
17
##   23   23   37   84  127  160  238  311  662  590  539  452  463  443  506
394
##   18   19   20
##  823  526 1599
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   11.00   15.00   14.68   19.00   20.00
##
## includes extended item information - examples:
##                                   labels
## 1 t_00007fba-6bd3-449d-85dd-54d4aea397c2
## 2 t_0000dc06-0c00-4a09-9dc6-3bdad9c6f0e8
## 3 t_00020dc1-1b82-43e9-8327-77b074bdf626
##
## includes extended transaction information - examples:
##                               transactionID
## 3357 0_04d229e5-f317-46de-bcec-f57667acd6eb
## 2774 0_0402175e-e336-48d7-8c2e-73ddfc554e46
## 6237 0_091e086c-f599-47a8-8386-3816837ea5d6
```

Responde a las siguientes preguntas:

- ¿Cuál es el ID y frecuencia absoluta de la sesión más frecuente? (**NOTE-26**): `t_bacf06d3-9185-4183-84ea-ff0db51475ce`, con 883 ocurrencias
- ¿Cuántos items tiene el itemset con mayor número de ocurrencias en la lista element length distribution? (**NOTE-27**): 20, con 1599 ocurrencias/transacciones
- En el contexto de este dataset con información de sesiones de streaming, ¿cuál es la interpretación de la media aritmética? (**NOTE-28**): el número promedio de canciones (o tracks) contenidos en una sesión de escucha, esto considerando que

estamos trabajando en el dataset y, donde agrupamos por ID de sesión las pistas (sus identificadores).

Ejecuta el algoritmo apriori con los parámetros por omisión.

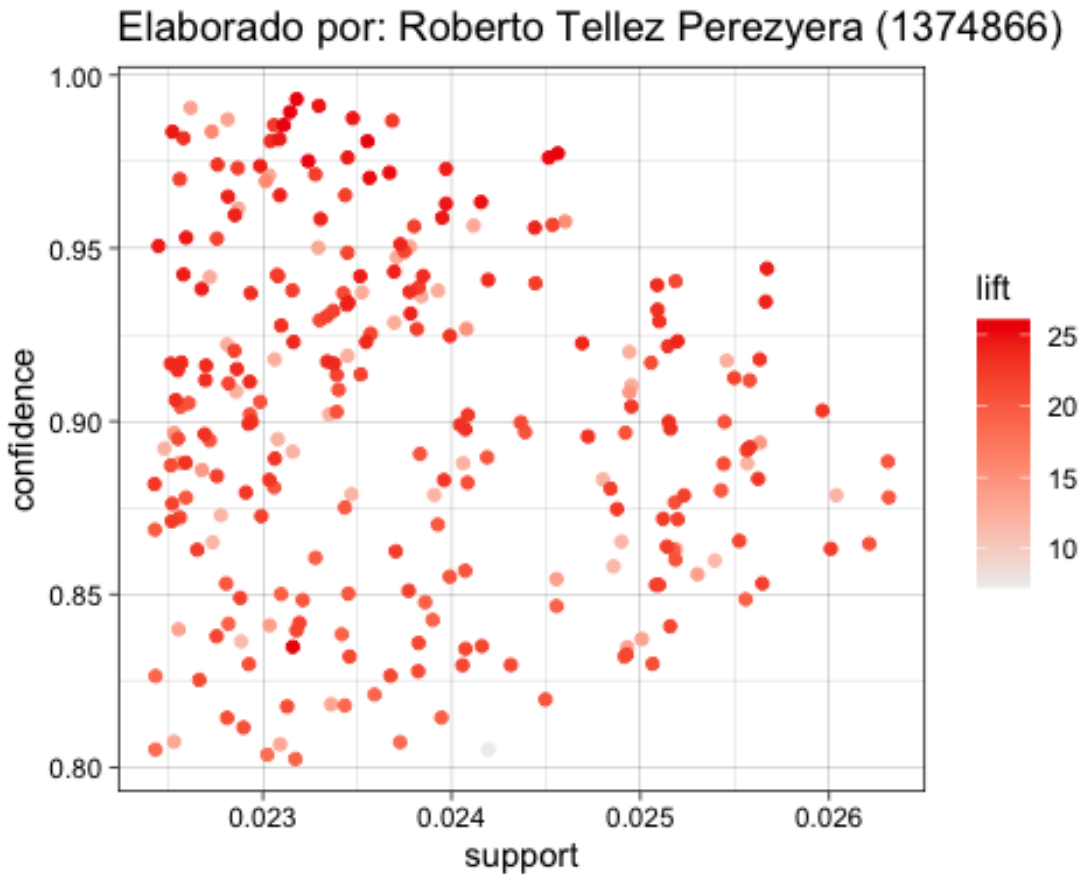
```
rules <- apriori(mssdtx)

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    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: 800
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[43526 item(s), 8000 transaction(s)] done [0.08s].
## sorting and recoding items ... [1 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 done [0.00s].
## writing ... [0 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].

summary(rules)

## set of 0 rules
```

- ¿Qué factores están asociados con que el algoritmo `apriori` con parámetros por omisión no haya producido ninguna regla de asociación, justifica tu respuesta? (**NOTE-29**): Pensemos en support y confidence, que son dos campos clave en la implementación del algoritmo apriori, pues definen qué reglas (de todo el conjunto posible) son consideradas o no según qué tan confiables o qué tanto support tienen.
- Se adjunta el identificador **NOTE-30** a continuación dado que no estaba presente.
- El últmo identificador más abajo (link a video) cambia de **NOTE-30** a **NOTE-31**
- ¿Qué riesgos identificas con la asignación de valores muy bajos en los parámetros de `support` y `confidence` al invocar el algoritmo `apriori`? (**NOTE-30**) Uno de los propósitos de apriori es generar reglas con un grado pertinente de solidez para tener pronósticos certeros en torno a transacciones y los items contenidos en ellas. Disminuir drásticamente `support` y `confidence` implicará generar reglas que no se cumplen con una frecuencia tan alta como otras, y también traerá como consecuencia el costo incrementado en tiempo y poder de cómputo que demanda generar más reglas (que no necesariamente serán de más pertinencia o utilidad).

CODE-07 Invoca `apriori` Configura la invocación a `apriori` para producir al menos 220 reglas de asociación para ello puedes considerar el uso de los parámetros `support` y `confidence` como se muestra a continuación `rules <- apriori(mssdtx, parameter = list(support = SUPP, confidence = CONF))`

```
#TODO: Configura las variables SUPP y CONF para generar al menos 250 reglas
de asociación

#SUPP = 0.1; CONF = 0.8   # initial combination generates 0 rules

# SUPP of 0.025 generates 156
# SUPP of 0.0125 generates 61309
# SUPP of 0.01875 with conf of 0.025 generates 2770 rules still in reasonable
time
# lowering SUPP to 0.02 brings 1508 rules
# SUPP of 0.0225 brings up 467 rules
# CONF in 0.025, bring it up to 0.3, generates 455 rules
# CONF brought back up to original value

# this combination generates 262 rules
SUPP = 0.0225
CONF = 0.8
rules <- apriori(mssdtx, parameter = list(support = SUPP, confidence = CONF))

## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.8    0.1    1 none FALSE            TRUE       5  0.0225      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: 180
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[43526 item(s), 8000 transaction(s)] done [0.07s].
## sorting and recoding items ... [48 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 done [0.00s].
## writing ... [262 rule(s)] done [0.00s].
## creating S4 object  ... done [0.01s].

summary(rules)

## set of 262 rules
##
```

```
## rule length distribution (lhs + rhs):sizes
##   2   3   4   5
##   1 163  88  10
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   3.000   3.000   3.408   4.000   5.000
##
## summary of quality measures:
##     support          confidence        coverage            lift
##  Min.   :0.02250   Min.   :0.8000   Min.   :0.02287   Min.   : 7.286
##  1st Qu.:0.02300   1st Qu.:0.8627   1st Qu.:0.02503   1st Qu.:19.275
##  Median :0.02350   Median :0.9009   Median :0.02600   Median :21.652
##  Mean   :0.02377   Mean   :0.9007   Mean   :0.02648   Mean   :20.193
##  3rd Qu.:0.02450   3rd Qu.:0.9375   3rd Qu.:0.02800   3rd Qu.:23.186
##  Max.   :0.02625   Max.   :0.9894   Max.   :0.03037   Max.   :26.058
##      count
##  Min.   :180.0
##  1st Qu.:184.0
##  Median :188.0
##  Mean   :190.2
##  3rd Qu.:196.0
##  Max.   :210.0
##
## mining info:
##    data ntransactions support confidence
##  mssdtx          8000  0.0225        0.8

try(
  plot(rules, main=paste0("Elaborado por: ", NOMBRE_COMPLETO, " (",
DIGITOS_MATRICULA, ")" ))
)

## To reduce overplotting, jitter is added! Use jitter = 0 to prevent jitter.
```



Inspecciona las reglas con mayor lift

```
inspect(head(sort(rules, by = "lift")))
```

```
##     lhs                                                rhs
support confidence coverage     lift count
## [1] {t_9a03f300-6504-4c7a-92b1-4cd88354ee74} => {t_e8fa3463-7ef7-4ca5-
8e65-2df979012c34} 0.023125  0.8371041 0.027625 26.05771   185
## [2] {t_1dba18ad-1ddb-4e97-983b-4af57e18d84a,
##      t_5718ab08-3a15-4d3f-9e63-42b2f6805e31,
##      t_7533e7ff-568e-448b-904e-fc0c3a9ec87e,
##      t_81b9a202-281e-42a4-b068-e0c0f44e1e4d} => {t_312b3509-48c1-4928-
a6e5-d3e7a51c216b} 0.023250  0.9893617 0.023500 25.61454   186
## [3] {t_1dba18ad-1ddb-4e97-983b-4af57e18d84a,
##      t_7533e7ff-568e-448b-904e-fc0c3a9ec87e,
##      t_81b9a202-281e-42a4-b068-e0c0f44e1e4d,
##      t_a0fa9956-ffc3-41c9-a344-e247baa0e321} => {t_312b3509-48c1-4928-
a6e5-d3e7a51c216b} 0.023125  0.9893048 0.023375 25.61307   185
## [4] {t_312b3509-48c1-4928-a6e5-d3e7a51c216b,
##      t_7533e7ff-568e-448b-904e-fc0c3a9ec87e,
##      t_81b9a202-281e-42a4-b068-e0c0f44e1e4d,
##      t_a0fa9956-ffc3-41c9-a344-e247baa0e321} => {t_1dba18ad-1ddb-4e97-
983b-4af57e18d84a} 0.023125  0.9893048 0.023375 25.53045   185
## [5] {t_1dba18ad-1ddb-4e97-983b-4af57e18d84a,
```

```
##      t_5718ab08-3a15-4d3f-9e63-42b2f6805e31,
##      t_7533e7ff-568e-448b-904e-fc0c3a9ec87e} => {t_312b3509-48c1-4928-
a6e5-d3e7a51c216b} 0.023500  0.9842932 0.023875 25.48332   188
## [6] {t_312b3509-48c1-4928-a6e5-d3e7a51c216b,
##      t_5718ab08-3a15-4d3f-9e63-42b2f6805e31,
##      t_7533e7ff-568e-448b-904e-fc0c3a9ec87e,
##      t_81b9a202-281e-42a4-b068-e0c0f44e1e4d} => {t_1dba18ad-1ddb-4e97-
983b-4af57e18d84a} 0.023250  0.9789474 0.023750 25.26316   186
```

Graba y publica un video (puede ser un video publicado de forma privada en YouTube u otro servicio similar) de 3 a 5 minutos explicando lo siguiente:

- Nombre y carrera.
- Concepto de association rules.
- Diferencia entre los conceptos de transacciones y reglas.
- Estrategia seguida para descubrir las reglas de interés en el dataset MSSD de este proyecto.
- Interpretación de la primera regla listada en el chunk anterior con el código `inspect(head(sort(rules, by = "lift")))` aclarando qué significan los datos listados en lhs, rhs y la interpretación de todas sus métricas.
- Provee el enlace al video aquí: **NOTE-31** https://youtu.be/zj2Fgrc2eDc

Entrega del proyecto final

- Verifica que el notebook corra de principio a fin sin errores y produzca los resultados esperados.
- Verifica que hayas realizado todas las actividades de programación indicadas con **CODE-nn**
- Verifica que hayas respondido a todas las preguntas **NOTE-nn**
- Verifica que hayas producido, publicado y escrito el enlace al video en el notebook.
- Guarda el archivo de markup (rmd) de forma local.
- Genera el PDF del notebook por medio del comando Knit (ya sea directo a PDF o Word y posteriormente Save As/PDF).
- No se recibirán proyectos en formato PDF con texto libre y capturas de pantalla.
- Publica el código de tu notebook a tu respositorio de GitHub por medio de los comandos `commit` y `push`.
- Envía el documento PDF por medio de mensaje personal a @marciano en Slack `naylacommunity` junto con el enlace a tu repositorio de GitHub.

Fecha límite de entrega: Sábado 5 de junio al cierre del día.