Proyecto Netflix Movies MADM

Laura Moreno, Josep Roman, Paul Ramírez

11/28/2020

Data Wrangle

Importación de datos

Info de los archivos "combined_data_.txt" The first line of each file contains the movie id followed by a colon. Each subsequent line in the file corresponds to a rating from a customer and its date in the following format:

CustomerID,Rating,Date

- MovieIDs range from 1 to 17770 sequentially.
- CustomerIDs range from 1 to 2649429, with gaps. There are 480189 users.
- Ratings are on a five star (integral) scale from 1 to 5.
- Dates have the format YYYY-MM-DD.

```
aux=read_tsv("Raw data/combined_data_1.txt", col_names = FALSE, n_max = 30000) #lectura de las primeras
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_character()
## )
```

Orden de los datos

```
aux=aux%>% mutate(fila=row_number()) #añadir columma con número de fila
filas=grep(":",aux$X1) #buscar filas con ":", filas comienzo nueva pelicula
filas_ID= aux %>% filter( fila %in% filas )
IDs=unique(filas_ID$X1)
reps=diff(c(filas_ID$fila,max(aux$fila)+1))
length(reps)
```

```
## [1] 17
```

```
dim(aux)
```

```
## [1] 30000     2
```

```r
sum(reps)
```

```
## [1] 30000
```

```r
scores = aux %>% mutate(ID1=rep(filas_ID$X1,times=reps)) %>% filter(!(fila %in% filas) )

#ahora borramos los datos de la última película por si se han cortado a medias
scores = scores %>% filter( scores$fila < filas_ID$fila[length(filas_ID$fila)-1] )

# Ahora arregloamos la variable X1, y separamos la fecha en año, mes y día
scores = scores %>% separate(X1,into = c("CustomerID","Score","Date"), sep = ",")
scores = scores %>% mutate(Date_copy = Date)  %>% separate(Date_copy, into = c("Year", "Month", "Day"),

#Renombramos y reordenamos las variables
scores <- rename(scores, MovieID = ID1, RowID = fila)
scores = scores %>% relocate(RowID, MovieID, CustomerID, Date, Year, Month, Day, Score)

#Quitamos los ":" de el campo MovieID
scores$MovieID <- scores$MovieID %>% str_replace(":", "")

# Cambiamos los tipos de variable necesarios

scores <- scores %>% mutate(across(c(RowID:CustomerID, Year:Score), as.integer))


summary(scores)
```

```
##      RowID          MovieID         CustomerID          Date
##  Min.   :    2   Min.   : 1.000   Min.   :      7   Length:21629
##  1st Qu.: 5416   1st Qu.: 8.000   1st Qu.: 666743   Class :character
##  Median :10823   Median : 8.000   Median :1339769   Mode  :character
##  Mean   :10822   Mean   : 7.346   Mean   :1332827
##  3rd Qu.:16230   3rd Qu.: 8.000   3rd Qu.:1994322
##  Max.   :21644   Max.   :15.000   Max.   :2649336
##       Year          Month             Day           Score
##  Min.   :2000   Min.   : 1.000   Min.   : 1.0   Min.   :1.000
##  1st Qu.:2005   1st Qu.: 4.000   1st Qu.: 8.0   1st Qu.:2.000
##  Median :2005   Median : 6.000   Median :16.0   Median :3.000
##  Mean   :2005   Mean   : 6.305   Mean   :15.7   Mean   :3.284
##  3rd Qu.:2005   3rd Qu.: 8.000   3rd Qu.:23.0   3rd Qu.:4.000
##  Max.   :2005   Max.   :12.000   Max.   :31.0   Max.   :5.000
```

Info data

```r
length(unique(scores$CustomerID)) #9619 usuarios distintos
```

```
## [1] 20537
```

```r
table(scores$Score) # frecuencia puntuaciones
```

```
##
##    1    2    3    4    5
## 2702 3157 5532 5765 4473
```

```
table(scores$MovieID) # frecuencia pelis
```

```
##
##     1     2     3     4     5     6     7     8     9    10    11    12    13
##   547   145  2012   142  1140  1019    93 14910    95   249   198   546   125
##    14    15
##   118   290
```

```
scores<-mutate(scores, fila=NULL) # eliminamos la columna fila
```

Cargar csv títulos

```
library(readr)
movie_titles <- read_csv("Raw data/movie_titles.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   MovieID = col_double(),
##   ReleaseDate = col_double(),
##   MovieTitle = col_character()
## )

## Warning: 357 parsing failures.
## row col  expected    actual                         file
##  72  -- 3 columns 4 columns 'Raw data/movie_titles.csv'
## 264  -- 3 columns 5 columns 'Raw data/movie_titles.csv'
## 350  -- 3 columns 4 columns 'Raw data/movie_titles.csv'
## 366  -- 3 columns 4 columns 'Raw data/movie_titles.csv'
## 394  -- 3 columns 4 columns 'Raw data/movie_titles.csv'
## ... ... ......... ......... ...........................
## See problems(...) for more details.
```

Para incorporar la columna a la tabla 'Scores' lo vamos a hacer mediante un `left_join`.

- El `left_join` se queda con todas las observaciones que aparecen en el primer dataset, es decir, solo tendrá en cuenta las películas que observadas en el primer dataset.
- El `join` entre tablas lo hemos hecho con la columna `MovieID`, presente en ambas tablas. Tal y como vemos en la tabla `movies_titles.csv`, cada película tiene un `MovieID` único, lo que se conoce como *clave primaria*. No obstante, en la tabla `scores` cada MovieID puede ser puntuada por varios `CustomerID`, en este caso, la *clave primaria* se constituye a partir de la combinación de ambas variables.

```
(scores %>%
  left_join(movie_titles,by = 'MovieID' ) -> scores)
```

```
## # A tibble: 21,629 x 10
##    RowID MovieID CustomerID Date   Year Month   Day Score ReleaseDate MovieTitle
##    <int>   <dbl>      <int> <chr> <int> <int> <int> <int>       <dbl> <chr>
##  1     2       1    1488844 2005~  2005     9     6     3        2003 Dinosaur ~
##  2     3       1     822109 2005~  2005     5    13     5        2003 Dinosaur ~
##  3     4       1     885013 2005~  2005    10    19     4        2003 Dinosaur ~
##  4     5       1      30878 2005~  2005    12    26     4        2003 Dinosaur ~
##  5     6       1     823519 2004~  2004     5     3     3        2003 Dinosaur ~
##  6     7       1     893988 2005~  2005    11    17     3        2003 Dinosaur ~
##  7     8       1     124105 2004~  2004     8     5     4        2003 Dinosaur ~
##  8     9       1    1248029 2004~  2004     4    22     3        2003 Dinosaur ~
##  9    10       1    1842128 2004~  2004     5     9     4        2003 Dinosaur ~
## 10    11       1    2238063 2005~  2005     5    11     3        2003 Dinosaur ~
## # ... with 21,619 more rows
```