

# Proyecto Netflix Movies MADM

Laura Moreno, Josep Roman, Paul Ramírez

11/28/2020

## Contenidos

<b>1</b>	<b>Objetivo</b>	<b>1</b>
<b>2</b>	<b>Data Wrangle</b>	<b>1</b>
2.1	Importación de datos . . . . .	1
2.2	Limpieza de los datos . . . . .	2
<b>3</b>	<b>Estadística Descriptiva</b>	<b>5</b>
<b>4</b>	<b>Sistema de Recomendación / Similaridad (opcional)</b>	<b>8</b>

## 1 Objetivo

## 2 Data Wrangle

### 2.1 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.

### Carga archivo puntuaciones películas

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

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

## Carga archivo títulos películas

```
rm(titles,tt)
```

```
## Warning in rm(titles, tt): object 'titles' not found
```

```
## Warning in rm(titles, tt): object 'tt' not found
```

```
titles = read_csv(here("Raw data",'movie_titles.csv'), col_names=F)
```

```
##
```

```
## -- Column specification -----
```

```
## cols(
```

```
##   X1 = col_character(),
```

```
##   X2 = col_character(),
```

```
##   X3 = col_character()
```

```
## )
```

```
## Warning: 343 parsing failures.
```

```
## row col expected actual
```

```
## 73 -- 3 columns 4 columns 'C:/Developer MADM/Cursos MADM Github/proyecto-netflix-movies-madm/Raw d
```

```
## 265 -- 3 columns 5 columns 'C:/Developer MADM/Cursos MADM Github/proyecto-netflix-movies-madm/Raw d
```

```
## 351 -- 3 columns 4 columns 'C:/Developer MADM/Cursos MADM Github/proyecto-netflix-movies-madm/Raw d
```

```
## 367 -- 3 columns 4 columns 'C:/Developer MADM/Cursos MADM Github/proyecto-netflix-movies-madm/Raw d
```

```
## 395 -- 3 columns 4 columns 'C:/Developer MADM/Cursos MADM Github/proyecto-netflix-movies-madm/Raw d
```

```
## ... ..
```

```
## See problems(...) for more details.
```

```
titles <- tibble(titles)
```

## 2.2 Limpieza de los datos

Limpieza datos puntuaciones películas

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

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

```
dim(aux)
```

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

```
sum(reps)
```

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

```
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 %<>% filter( scores$fila < filas_ID$fila[length(filas_ID$fila)-1] )
```

```
# Ahora arreglamos la variable X1, y separamos la fecha en año, mes y día
```

```
scores %<>% separate(X1,into = c("CustomerID","Score","Date"), sep = ",")
```

```
scores %<>% mutate(Date_copy = Date) %>% separate(Date_copy, into = c("Year", "Month", "Day"), sep = ",")
```

```
#Renombramos y reordenamos las variables
```

```
scores %<>% rename(MovieID = ID1)
```

```
scores <- select(scores, -fila) # eliminamos la columna fila
```

```
scores %<>% relocate(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 %<>% mutate(across(c(MovieID:CustomerID, Year:Score), as.integer))
```

```
scores %<>% mutate(Date = as.Date(Date))
```

```
summary(scores)
```

```
##      MovieID      CustomerID      Date      Year  
## Min.   : 1.000   Min.    :      7   Min.    :2000-01-13   Min.    :2000  
## 1st Qu.: 8.000   1st Qu.: 666743   1st Qu.:2005-03-28   1st Qu.:2005  
## Median : 8.000   Median :1339769   Median :2005-05-17   Median :2005  
## Mean   : 7.346   Mean    :1332827   Mean    :2005-04-03   Mean    :2005  
## 3rd Qu.: 8.000   3rd Qu.:1994322   3rd Qu.:2005-08-02   3rd Qu.:2005  
## Max.   :15.000   Max.    :2649336   Max.    :2005-12-31   Max.    :2005  
##      Month      Day      Score  
## Min.   : 1.000   Min.    : 1.0   Min.    :1.000  
## 1st Qu.: 4.000   1st Qu.: 8.0   1st Qu.:2.000  
## Median : 6.000   Median :16.0   Median :3.000  
## Mean   : 6.305   Mean    :15.7   Mean    :3.284  
## 3rd Qu.: 8.000   3rd Qu.:23.0   3rd Qu.:4.000  
## Max.   :12.000   Max.    :31.0   Max.    :5.000
```

Vemos que tenemos información de la películas 1 a la 15, y las puntuaciones se hicieron entre el 2000 y el 2005 (mayoritariamente en 2005). Distribución de los meses y días en que se puntuo es uniforme.

Veamos más información sobre los datos:

```
length(unique(scores$CustomerID)) #20537 usuarios distintos
```

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

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

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

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

```
##
##      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
```

Limpieza datos títulos películas

```
head(titles)
```

```
## # A tibble: 6 x 3
##   X1      X2      X3
##   <chr> <chr> <chr>
## 1 MovieID ReleaseDate MovieTitle
## 2 1      2003      Dinosaur Planet
## 3 2      2004      Isle of Man TT 2004 Review
## 4 3      1997      Character
## 5 4      1994      Paula Abdul's Get Up & Dance
## 6 5      2004      The Rise and Fall of ECW
```

```
titles %<>% rename(MovieID = X1, Release_Year = X2, Title = X3)
titles %<>% mutate(across(c(MovieID:Release_Year), as.integer))
```

```
## Warning: Problem with 'mutate()' input '...1'.
## i NAs introduced by coercion
## i Input '...1' is 'across(c(MovieID:Release_Year), as.integer)'.
```

```
## Warning in fn(col, ...): NAs introduced by coercion
```

```
## Warning: Problem with 'mutate()' input '...1'.
## i NAs introduced by coercion
## i Input '...1' is 'across(c(MovieID:Release_Year), as.integer)'.
```

```
## Warning in fn(col, ...): NAs introduced by coercion
```

**\*\*Left join de puntuaciones películas con los títulos**

Hacemos un left join con 'titles' para añadir a la tabla 'scores' los títulos de cada película y el año en que se publicaron

- 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(titles, by = 'MovieID')
summary(scores); head(scores)
```

```
##      MovieID      CustomerID      Date      Year
## Min.   : 1.000   Min.      : 7   Min.   :2000-01-13   Min.   :2000
## 1st Qu.: 8.000   1st Qu.: 666743   1st Qu.:2005-03-28   1st Qu.:2005
## Median : 8.000   Median :1339769   Median :2005-05-17   Median :2005
## Mean   : 7.346   Mean   :1332827   Mean   :2005-04-03   Mean   :2005
## 3rd Qu.: 8.000   3rd Qu.:1994322   3rd Qu.:2005-08-02   3rd Qu.:2005
## Max.   :15.000   Max.   :2649336   Max.   :2005-12-31   Max.   :2005
##      Month      Day      Score      Release_Year
## Min.   : 1.000   Min.   : 1.0   Min.   :1.000   Min.   :1947
## 1st Qu.: 4.000   1st Qu.: 8.0   1st Qu.:2.000   1st Qu.:2003
## Median : 6.000   Median :16.0   Median :3.000   Median :2004
## Mean   : 6.305   Mean   :15.7   Mean   :3.284   Mean   :2001
## 3rd Qu.: 8.000   3rd Qu.:23.0   3rd Qu.:4.000   3rd Qu.:2004
## Max.   :12.000   Max.   :31.0   Max.   :5.000   Max.   :2004
##      Title
## Length:21629
## Class :character
## Mode  :character
##
##
##
```

```
## # A tibble: 6 x 9
##      MovieID CustomerID Date      Year Month   Day Score Release_Year Title
##      <int>      <int> <date>    <int> <int> <int> <int>      <int> <chr>
## 1         1    1488844 2005-09-06 2005     9     6     3      2003 Dinosaur P~
## 2         1     822109 2005-05-13 2005     5    13     5      2003 Dinosaur P~
## 3         1     885013 2005-10-19 2005    10    19     4      2003 Dinosaur P~
## 4         1      30878 2005-12-26 2005    12    26     4      2003 Dinosaur P~
## 5         1     823519 2004-05-03 2004     5     3     3      2003 Dinosaur P~
## 6         1     893988 2005-11-17 2005    11    17     3      2003 Dinosaur P~
```

### 3 Estadística Descriptiva

1. Justifica para cada una de las variables de la tabla anterior el tipo de dato que mejor se ajusta a cada una de ellas: numérico, ordinal, categórico. . . .
2. Estudia la distribución del numero de películas estrenadas por año. Realiza un gráfico de muestre esta distribución haciendo los ajustes necesarios (agrupaciones, cambios de escala, transformaciones. . . )
3. Investiga la librería lubridate (o la que consideréis para manipulación de datos) y utilízala para transformar la columna de la fecha de la valoración en varias columnas por ejemplo year, month, week, day\_of\_week.

4. Genera un tabla que para cada película nos dé el número total de valoraciones, la suma de las valoraciones, la media las valoraciones, y otras estadísticos de interés (desviación típica, moda , mediana).
5. De las cinco películas con más número total de valoraciones, compara sus estadísticos y distribuciones (histogramas, boxplot, violin plot,. . . )
6. Investiga la distribución de valoraciones por día de la semana y por mes.¿Qué meses y días de la semana se valoran más películas en netflix?
7. Genera una tabla agrupada por película y año del número de valoraciones. Representa la tabla gráficamente para de las 10 películas con mayor número de valoraciones .
8. Distribución del score promedio por año de las 10 películas con mayor número de valoraciones.
9. Realiza algún gráfico o estudio de estadísticos adicional que consideres informativo en base al análisis exploratorio anterior.
  1. Puntuaciones por fecha
  2. Puntuaciones por película
  3. Puntuaciones por usuario
  4. Número de puntuaciones por película, usuario y año lanzamiento
  5. Distribucion de los scores (boxplot,barplot)
  6. Series temporales de puntuaciones
  7. Distribución de cuantos usuarios evaluan cuantas pelis totales y diferentes

Valoración media por película, de mayor a menor:

```
movie_score_avg <- scores %>%
  group_by(MovieID) %>%
  summarise(Mean_Score = mean(Score), n = n()) %>%
  left_join(titles, by = "MovieID") %>%
  arrange(desc(Mean_Score))
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
movie_score_avg
```

```
## # A tibble: 15 x 5
##   MovieID Mean_Score      n Release_Year Title
##   <int>     <dbl> <int>      <int> <chr>
## 1      13      4.55   125        2003 Lord of the Rings: The Return of the K-
## 2       5      3.92  1140        2004 The Rise and Fall of ECW
## 3       1      3.75   547        2003 Dinosaur Planet
## 4       3      3.64  2012        1997 Character
## 5       2      3.56   145        2004 Isle of Man TT 2004 Review
## 6      12      3.42   546        1947 My Favorite Brunette
## 7      15      3.29   290        1988 Neil Diamond: Greatest Hits Live
## 8       8      3.19 14910        2004 What the $*! Do We Know!?
## 9      10      3.18   249        2001 Fighter
## 10      6      3.08  1019        1997 Sick
## 11     11      3.03   198        1999 Full Frame: Documentary Shorts
## 12     14      3.03   118        1982 Nature: Antarctica
## 13      4      2.74   142        1994 Paula Abdul's Get Up & Dance
## 14      9      2.62    95        1991 Class of Nuke 'Em High 2
## 15      7      2.13    93        1992 8 Man
```

Valoración media por 'Release\_Year', de mayor a menor:

```
release_year_score_avg <- scores %>%  
  group_by(Release_Year) %>%  
  summarise(Mean_Score = mean(Score), n = n()) %>%  
  arrange(desc(Mean_Score))
```

## 'summarise()' ungrouping output (override with '.groups' argument)

```
release_year_score_avg
```

```
## # A tibble: 11 x 3  
##   Release_Year Mean_Score      n  
##   <int>      <dbl> <int>  
## 1     2003      3.90   672  
## 2     1997      3.45  3031  
## 3     1947      3.42   546  
## 4     1988      3.29   290  
## 5     2004      3.24 16195  
## 6     2001      3.18   249  
## 7     1999      3.03   198  
## 8     1982      3.03   118  
## 9     1994      2.74   142  
## 10    1991      2.62    95  
## 11    1992      2.13    93
```

Valoración media por día de la semana, de mayor a menor:

```
scores_day_week <- scores %>% mutate(Day_Week = weekdays(Date))  
scores_day_week %<>% mutate(Is_Weekend = isWeekend(Date))  
  
day_week_score_avg <- scores_day_week %>%  
  group_by(Day_Week) %>%  
  summarise(Mean_Score = mean(Score), n = n()) %>%  
  arrange(desc(Mean_Score))
```

## 'summarise()' ungrouping output (override with '.groups' argument)

```
day_week_score_avg
```

```
## # A tibble: 7 x 3  
##   Day_Week Mean_Score      n  
##   <chr>      <dbl> <int>  
## 1 Friday      3.32  2767  
## 2 Thursday     3.31  2974  
## 3 Saturday     3.31  1851  
## 4 Tuesday      3.29  4094  
## 5 Monday       3.29  4230  
## 6 Wednesday    3.26  3537  
## 7 Sunday       3.22  2176
```

Valoración media entre semana / fin de semana:

```
weekend_weekday_score_avg <- scores_day_week %>%  
  group_by(Is_Weekend) %>%  
  summarise(Mean_Score = mean(Score), n = n())
```

```
## 'summarise()' ungrouping output (override with '.groups' argument)
```

```
weekend_weekday_score_avg
```

```
## # A tibble: 2 x 3  
##   Is_Weekend Mean_Score      n  
##   <lgl>         <dbl> <int>  
## 1 FALSE          3.29 17602  
## 2 TRUE           3.26  4027
```

```
n_scores_weekend = weekend_weekday_score_avg %>% filter(Is_Weekend == TRUE) %>% select(n)
```

```
n_scores = sum(weekend_weekday_score_avg$n)
```

```
n_scores_weekend_weekday_ratio = n_scores_weekend / n_scores #el 18% de las valoraciones son en fin de semana
```

## 4 Sistema de Recomendación / Similaridad (opcional)