Proyecto Netflix Movies MADM

Laura Moreno, Josep Roman, Paul Ramírez

11/28/2020

Contenidos

1 Objetivo

2 Data Wrangle

2.1 Importación de datos

2.1.1 Importación datos puntuaciones películas

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
```

2.1.2 Importación datos información sobre las películas

Carga archivo titulos películas

```
rm(titles,tt)
#algunas peliculas tienen una coma en su nombre, así que cargamos primero todo como una única columna,
titles = read_table(here("Raw data",'movie_titles.csv')) %>%
  separate(col = 1, into = c("MovieID", "Release_Year", "Title"), sep = ",", extra = "merge")
```

2.2 Limpieza de los datos

2.2.1 Limpieza datos puntuaciones películas

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

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

```
dim(aux)
```

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

```
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 arregloamos 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
```

```r
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 peliculas 1 a la 15, y las puntuaciones se hicieron entre el 2000 y el 2005 (mayoritariamente en 2005). Distribución de los meses y dias en que se puntuo es uniforme.

Veamos más informacion sobre los datos:

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

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

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

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

```r
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
```

2.2.2 Limpieza datos títulos películas

```r
head(titles)
```

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

```
titles  %<>% mutate(across(c(MovieID:Release_Year), as.integer))
```

**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')
kable(summary(scores))
```

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

```
kable(head(scores))
```

MovieID	CustomerID	Date	Year	Month	Day	Score	Release_Year	Title
1	1488844	2005-09-06	2005	9	6	3	2003	Dinosaur Planet
1	822109	2005-05-13	2005	5	13	5	2003	Dinosaur Planet
1	885013	2005-10-19	2005	10	19	4	2003	Dinosaur Planet
1	30878	2005-12-26	2005	12	26	4	2003	Dinosaur Planet
1	823519	2004-05-03	2004	5	3	3	2003	Dinosaur Planet
1	893988	2005-11-17	2005	11	17	3	2003	Dinosaur Planet

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. . . .

```
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        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   
##                     
##                     
##
```

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. . .)

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))

kable(release_year_score_avg)
```

Release_Year	Mean_Score	n
2003	3.898810	672
1997	3.453976	3031
1947	3.417582	546
1988	3.286207	290
2004	3.244458	16195
2001	3.180723	249
1999	3.030303	198
1982	3.025424	118
1994	2.739437	142
1991	2.621053	95
1992	2.129032	93

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.

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))

kable(day_week_score_avg)
```

Day_Week	Mean_Score	n
Friday	3.319118	2767
Thursday	3.313383	2974
Saturday	3.307942	1851
Tuesday	3.287494	4094
Monday	3.286998	4230
Wednesday	3.255584	3537
Sunday	3.215993	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())

kable(weekend_weekday_score_avg)
```

Is_Weekend	Mean_Score	n
FALSE	3.290308	17602
TRUE	3.258257	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
```

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).

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))

kable(movie_score_avg)
```

MovieID	Mean_Score	n	Release_Year	Title
13	4.552000	125	2003	Lord of the Rings: The Return of the King: Extended Edition: Bonus Material
5	3.919298	1140	2004	The Rise and Fall of ECW
1	3.749543	547	2003	Dinosaur Planet
3	3.641153	2012	1997	Character
2	3.558621	145	2004	Isle of Man TT 2004 Review
12	3.417582	546	1947	My Favorite Brunette
15	3.286207	290	1988	Neil Diamond: Greatest Hits Live
8	3.189805	14910	2004	What the #$*! Do We Know!?
10	3.180723	249	2001	Fighter
6	3.084396	1019	1997	Sick
11	3.030303	198	1999	Full Frame: Documentary Shorts
14	3.025424	118	1982	Nature: Antarctica
4	2.739437	142	1994	Paula Abdul's Get Up & Dance
9	2.621053	95	1991	Class of Nuke 'Em High 2
7	2.129032	93	1992	8 Man

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 estudió 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

4 Sistema de Recomendación / Similaridad (opcional)