Proyecto Netflix Movies MADM

Laura Moreno, Josep Roman, Paul Ramírez

11/28/2020

Contenidos

1 Objetivo **1**

2 Data Wrangle **1**

2.1 Importación de datos 1

2.2 Limpieza de los datos 2

3 Estadística Descriptiva **3**

4 Sistema de Recomendación / Similaridad (opcional) **6**

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.

2.1.2 Importación datos información sobre las películas

Carga archivo titulos películas

```
rm(titles)
#algunas peliculas tienen una coma en su nombre, así que cargamos primero todo como una única columma, para luego dividirlo en 3,
titles = read_table(here("Data",'movie_titles_raw.csv'), col_names=F) %>%
  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

```
scores = read_csv(here("Data","nuestras_pelis.csv"))

# 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.   :   60   Min.   :      6   Min.   :1999-11-11   Min.   :1999
##  1st Qu.: 3893   1st Qu.: 661465   1st Qu.:2004-04-14   1st Qu.:2004
##  Median : 7772   Median :1320113   Median :2005-01-10   Median :2005
##  Mean   : 7924   Mean   :1322801   Mean   :2004-09-27   Mean   :2004
##  3rd Qu.:10759   3rd Qu.:1984315   3rd Qu.:2005-07-05   3rd Qu.:2005
##  Max.   :17748   Max.   :2649429   Max.   :2005-12-31   Max.   :2005
##      Month             Day             Score
##  Min.   : 1.000   Min.   : 1.00   Min.   :1.000
##  1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:3.000
##  Median : 7.000   Median :16.00   Median :4.000
##  Mean   : 6.734   Mean   :15.72   Mean   :3.601
##  3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:4.000
##  Max.   :12.000   Max.   :31.00   Max.   :5.000
```

2.2.2 Limpieza datos títulos películas

```
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')
summary(scores)
```

```
##     MovieID        CustomerID           Date                Year
##  Min.   :   60   Min.   :      6   Min.   :1999-11-11   Min.   :1999
##  1st Qu.: 3893   1st Qu.: 661465   1st Qu.:2004-04-14   1st Qu.:2004
##  Median : 7772   Median :1320113   Median :2005-01-10   Median :2005
##  Mean   : 7924   Mean   :1322801   Mean   :2004-09-27   Mean   :2004
##  3rd Qu.:10759   3rd Qu.:1984315   3rd Qu.:2005-07-05   3rd Qu.:2005
##  Max.   :17748   Max.   :2649429   Max.   :2005-12-31   Max.   :2005
##      Month             Day            Score        Release_Year
##  Min.   : 1.000   Min.   : 1.00   Min.   :1.000   Min.   :1927
##  1st Qu.: 4.000   1st Qu.: 8.00   1st Qu.:3.000   1st Qu.:1987
##  Median : 7.000   Median :16.00   Median :4.000   Median :2000
##  Mean   : 6.734   Mean   :15.72   Mean   :3.601   Mean   :1993
##  3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:4.000   3rd Qu.:2002
##  Max.   :12.000   Max.   :31.00   Max.   :5.000   Max.   :2005
##     Title
##  Length:2629059
##  Class :character
##  Mode  :character
##
##
##
```

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

MovieID	CustomerID	Date	Year	Month	Day	Score	Release_Year	Title
515	2295232	2005-08-16	2005	8	16	1	2005	Avia Vampire Hunter
515	1560318	2005-10-04	2005	10	4	3	2005	Avia Vampire Hunter
515	2550394	2005-11-01	2005	11	1	1	2005	Avia Vampire Hunter
515	1502043	2005-08-15	2005	8	15	1	2005	Avia Vampire Hunter
515	1507284	2005-10-03	2005	10	3	1	2005	Avia Vampire Hunter
515	771626	2005-08-11	2005	8	11	1	2005	Avia Vampire Hunter

3 Estadística Descriptiva

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:

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

```
## [1] 378041
```

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

```
##
##      1      2      3      4      5
## 114760 261162 762713 909178 581246
```

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

```
##
## 515
##   6
```

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. . . .

```
glimpse(scores)
```

```
## Rows: 2,629,059
## Columns: 9
## $ MovieID      <int> 515, 515, 515, 515, 515, 515, 515, 515, 515, 515, 515,...
## $ CustomerID   <int> 2295232, 1560318, 2550394, 1502043, 1507284, 771626, 1...
## $ Date         <date> 2005-08-16, 2005-10-04, 2005-11-01, 2005-08-15, 2005-...
## $ Year         <int> 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, ...
## $ Month        <int> 8, 10, 11, 8, 10, 8, 9, 8, 6, 8, 8, 8, 7, 8, 7, 8, 8, ...
## $ Day          <int> 16, 4, 1, 15, 3, 11, 11, 6, 14, 10, 8, 5, 30, 2, 5, 27...
## $ Score        <int> 1, 3, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, ...
## $ Release_Year <int> 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, 2005, ...
## $ Title        <chr> "Avia Vampire Hunter", "Avia Vampire Hunter", "Avia Va...
```

3.0.1 Variables tipo *int*: MovieID, CustomerID, Score, Release_Year

- *CustomerID*: Contiene un número entero, estos son objetos que contienen un único campo, un identificado ID para cada cliente, no queremos duplicados.
- *MovieID*: Contiene un número entero, estos son objetos que contienen un único campo, un identificado ID para cada película, no queremos duplicados. Un integer es inmutable.
- *Release_Year*: No existen años con decimales, por lo tanto utilizar variables para datos enteros seria suficiente. Movie_title: chr. Utilizamos el tipo carácter porque nos interesan objetos que representan un conjunto de letras.
- *Score*: Las puntuaciones son números enteros del 1 al 5. Las películas no aceptan decimales como puntuación.

3.0.2 Variables tipo *date*: Date

- *Date* : esta variable incluye datos de tipo fecha (YY/MM/DD) por ello lo más adecuado es tratarlo como una variable de este tipo. Gracias a esto, podemos aplicar paquetes como *lubridate* para manipular fechas.

3.0.3 Variables tipo *chr*: Title

- *Title*: Utilizamos el tipo carácter porque nos interesan objetos que representan un conjunto de letras.

2. Estudia la distribución del número 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(head(release_year_score_avg))
```

Release_Year	Mean_Score	n
1964	4.117170	6290
1984	4.087852	57961
1937	4.014899	3893

Release_Year	Mean_Score	n
1934	3.964283	11619
1968	3.913478	12598
1994	3.890081	46216

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
viernes	3.616880	356366
domingo	3.609243	280412
sábado	3.604435	262885
miércoles	3.603150	436629
jueves	3.601141	378354
lunes	3.592429	451006
martes	3.590045	463407

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	3.599902	2085762
TRUE	3.606917	543297

```
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, que es menos que
```

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:

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

kable(head(movie_score_avg))
```

MovieID	Mean_Score	n	Release_Year	Title
14791	4.600000	75	2003	Trailer Park Boys: Season 3
16265	4.489491	5757	1977	Star Wars: Episode IV: A New Hope
14806	4.318058	7354	2001	Angel: Season 3
14283	4.299783	5524	1994	The Best of Friends: Vol. 3
16766	4.182796	372	1993	Inspector Morse 10: Deceived by Flight
14185	4.150541	6005	1964	Mary Poppins

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)