Proyecto Netflix Movies MADM

Laura Moreno, Josep Roman, Paul Ramírez

11/28/2020

Contenidos

1 Objetivo

Este documento presenta los resultados del análisis exploratorio realizado sobre valoraciones de usuarios de películas de *Neflix* *. En este informe se explora la distribución de las puntuaciones por distintos períodos de tiempo, se comparan los estadísticos de las películas que han recibido más votaciones y se analiza la distribución del número de votaciones por usuario.

*Datos extraidos del dataset "Neflix Prize Data" en Kaggle.

2 Data wrangle

2.1 Importación de datos

2.1.1 Importación datos puntuaciones películas

Creamos una semilla especifica para seleccionar aleatoriamente 250 películas con las que desarrollar el análisis exploratorio.

```
filas_ID_combined_all = read.csv(here("Data","filas_ID_combined_all.txt"))
set.seed(081034)
n_filas = nrow(filas_ID_combined_all)
muestra_grupo = sample(1:n_filas, 250, replace=F)
pelis <- filas_ID_combined_all[as.vector(muestra_grupo),]
```

Cargamos los 4 archivos originales con las puntuaciones, siguiendo el siguiente patrón:

```
data1 = read_tsv(here("Raw data","combined_data_1.txt"),col_names = FALSE)
```

Generamos un tibble vacío, y en función del archivo en el que se encuentre la película, vamos añadiendo en `scores` las filas correspondientes a nuestras películas:

```
scores = tibble()
for(i in 1:nrow(pelis)){
  if (data[i]==1){
    scores = rbind(scores,data1[fila[i]:fila_final[i],])
  }
  else if (data[i]==2){
    scores = rbind(scores,data2[fila[i]:fila_final[i],])
  }
  else if (data[i]==3){
    scores = rbind(scores,data3[fila[i]:fila_final[i],])
  }
  else {
    scores = rbind(scores,data4[fila[i]:fila_final[i],])
  }
}
```

Guardamos un csv con solo nuestras 250 películas en el formato original

```
write_csv(scores, here("Data", "nuestras_pelis_raw.csv"))
```

Cargamos el csv de nuestras 250 películas generado en el paso anterior:

```
aux = read_csv(here("Data", "nuestras_pelis_raw.csv"), col_names = T)
```

2.1.2 Importación datos títulos películas

```
titles = read_table(here("Data",'movie_titles_raw.csv'), col_names=F) %>%
  separate(col = 1, into = headers_titles, sep = ",", extra = "merge")
```

2.2 Preparación de los datos

2.2.1 Limpieza datos puntuaciones películas

Aplicamos el código de Ricardo para limpiar el dataframe `aux` y pasar al dataframe `scores` con una fila para cada valoración de usuario. A continuación, reorganizamos variables las variables:

```
scores %<>% relocate(MovieID, UserID, Date, Score)
```

2.2.2 Join de `scores` con `titles`

Hacemos un *left join* de `scores` con `titles` para añadir los títulos de cada película y el año de lanzamiento:

```
scores %<>% left_join(titles, by = 'MovieID')
```

2.2.3 Exportación e importación de datos limpios para su análisis

Datos puntuaciones películas

```
write_csv(scores,here("Data", "nuestras_pelis.csv"))
```

```
scores = read_csv(here("Data","nuestras_pelis.csv"))
scores %<>% mutate(across(c(MovieID,UserID,Score,Release_Year), as.integer), Date = as.Date(Date))
```

Datos títulos películas

```
write_delim(titles, here("Data", "nuestros_titles.csv"), delim = "|")
```

```
titles = read_delim(here("Data", "nuestros_titles.csv"), delim = "|")
titles  %<>% mutate(across(c(MovieID:Release_Year), as.integer))
```

Table 1: Daframe títulos películas

MovieID	Release_Year	Title
60	1969	The Libertine
65	2000	Lost in the Pershing Point Hotel
134	1996	Spirit Lost

3 Estadística descriptiva

3.1 Resumen

El dataset `scores` de valoraciones de Netflix contiene 1508892 valoraciones de películas, realizadas por 327577 usuarios diferentes para un catálogo de 250 películas con fecha de lanzamiento en Netflix desde 1927 hasta 2005. Las valoraciones han sido realizadas entre los años 1999 y 2005, usando una escala ordinal del 1 al 5.

Veamos más informacion sobre los datos:

3.2 Tipo de variables

```
glimpse(scores)
```

```
## Rows: 1,508,892
## Columns: 6
## $ MovieID      <int> 515, 515, 515, 515, 515, 515, 515, 515, 515, 515, 515,...
## $ UserID       <int> 2295232, 1560318, 2550394, 1502043, 1507284, 771626, 1...
## $ Date         <date> 2005-08-16, 2005-10-04, 2005-11-01, 2005-08-15, 2005-...
## $ 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...
```

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

- *UserID* & *MovieID* : Contiene un número entero, estos son objetos que contienen un único campo, un identificado ID para cada usuario (o película), no queremos duplicados. En cuanto al *MovieID*, será transformado en las gráficas a *chr* para visualizarlo mejor.
- *Release_Year*: No existen años con decimales, por lo tanto utilizar variables para datos enteros seria suficiente.
- *Score*: Las puntuaciones son números enteros del 1 - 5.

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.

Variables tipo *chr*: Title

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

3.3 Estadísticos dataframe puntuaciones

```
movie_scores <- scores %>%
  group_by(MovieID) %>%
  summarise(Sum_Score = sum(Score), Mean_Score = mean(Score), SD_Score = sd(Score),
            Mode_Score = mlv(Score), Median_Score = median(Score) , n = n()) %>%
  left_join(titles, by = 'MovieID')

movie_scores_table <- movie_scores %>%
  ungroup() %>%
  select(-MovieID, -Release_Year) %>%
  relocate(Title, n, Sum_Score, Mean_Score, SD_Score, Mode_Score, Mean_Score)
```

Table 2: Estadísticos puntuaciones

Title	n	Sum_Score	Mean_Score	SD_Score	Mode_Score	Median_Score
Curb Your Enthusiasm: Season 3	12148	52674	4.336	1.000	5	5
Prime Suspect 3	2637	11222	4.256	0.899	5	4
Singin' in the Rain	29225	119852	4.101	0.947	5	4

3.4 Transformación variable fecha valoración

Usamos la librería *lubridate* para generar variables separadas para año, número de mes, número de semana del año, número de día del mes, número de día de la semana, y una variable binaria que specifica si el día es fin de semana o entre semana.

```
scores_dates <- scores %>%
  mutate(
    Year = year(Date),
    n_month = month(Date),
    Week = week(Date),
    Day = day(Date),
    n_day_week = wday(Date, week_start = getOption("lubridate.week.start", 1)),
    Is_Weekend = if_else( isWeekend(Date) == TRUE, "Weekend", "Weekday" )
  )
```

A partir de las variables de número de mes y número de día de la semana, creamos dos factores ordenados para el mes y el día de la semana.

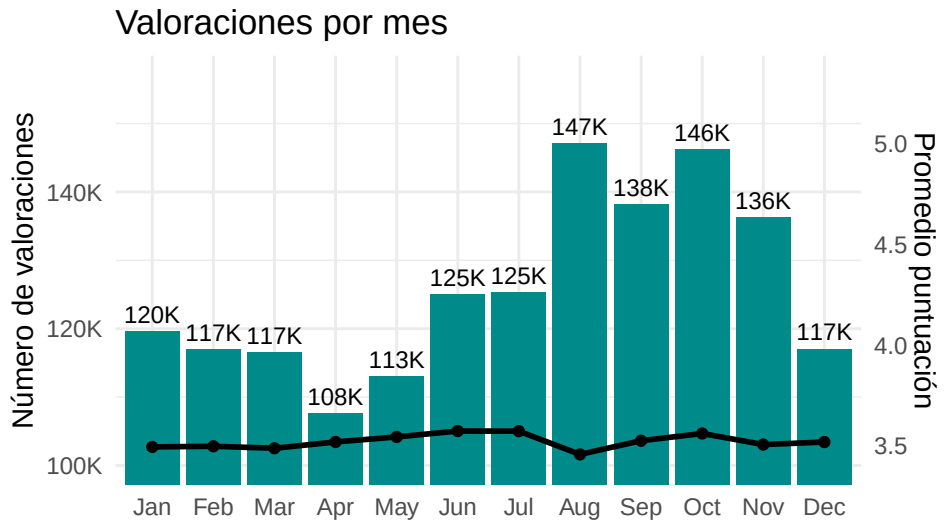
```
scores_dates %<>% mutate(
  Month = ordered(n_month, levels = seq(1, 12, 1),labels = month.abb),
  Day_Week = ordered(n_day_week, levels = seq(1, 7, 1), labels = day.abb)
)

scores_dates_table <- scores_dates %>%
  select(MovieID, UserID, Score, Date, Year, Month, Day, Day_Week, Is_Weekend)
```

Table 3: Dataframe puntuaciones con detalle fecha

MovieID	UserID	Score	Date	Year	Month	Day	Day_Week	Is_Weekend
9003	510180	3	1999-11-11	1999	Nov	11	Thu	Weekday
3893	122223	3	1999-12-08	1999	Dec	8	Wed	Weekday
6928	204439	3	1999-12-09	1999	Dec	9	Thu	Weekday

3.5 Análisis del número de valoraciones por més y día de la semana

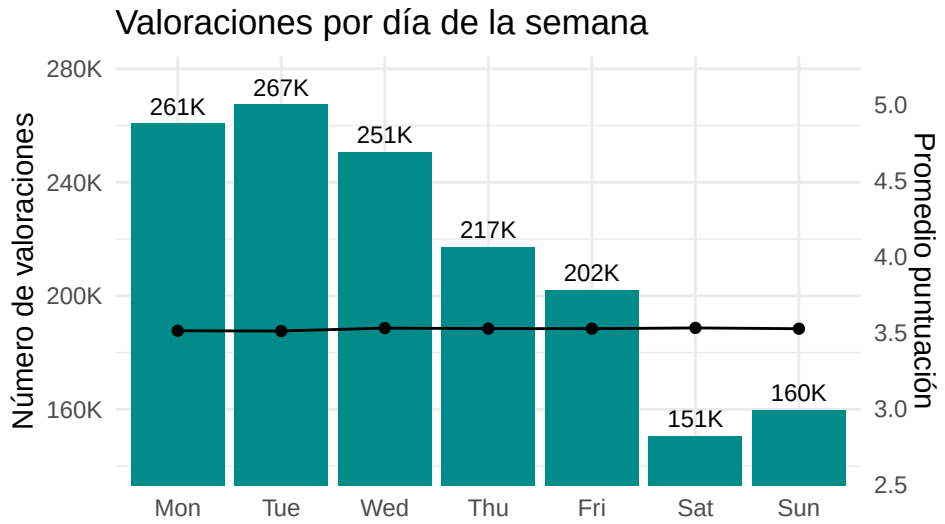
```
ggplot(data = month_scores, aes(x = Month)) +
  geom_bar(aes(y = n), fill = "darkcyan", stat = "identity") +
  coord_cartesian(ylim = c(n_min_limit, n_max_limit)) +
  geom_point(aes(y = Mean_Score/coeff)) +
  geom_line(aes(y = Mean_Score/coeff), size = 1, group = 1) +
  scale_y_continuous(
    name = "Número de valoraciones",
    labels = ks,
    sec.axis = sec_axis(~.*coeff, name = "Promedio puntuación")
  ) +
  labs(title = "Valoraciones por mes", x = "") +
  geom_text(aes(y = n, label = ks(n)), angle = 0, vjust = -0.5, size = 3) +
  theme_minimal()
```



En este primer gráfico se puede observar que hay una actividad considerablemente mayor de actividad finales de verano y durante el otoño. Sin embargo, la puntuación media de las valoraciones apenas fluctua y se situa en torno al 3.5.

```
ggplot(data = day_week_scores, aes(x = Day_Week)) +
  geom_bar(aes(y = n), fill = "darkcyan", stat = "identity") +
  coord_cartesian(ylim = c(n_min_limit, n_max_limit)) +
  geom_point(aes(y = Mean_Score/coeff)) +
  geom_line(aes(y = Mean_Score/coeff), group = 1) +
  scale_y_continuous(
```

```
    name = "Número de valoraciones",
    labels = ks,
    sec.axis = sec_axis(~.*coeff, name = "Promedio puntuación")
  ) +
  labs(title = "Valoraciones por día de la semana", x = "") +
  geom_text(aes(y = n, label = ks(n)), angle = 0, vjust = -0.5, size = 3) +
  theme_minimal()
```

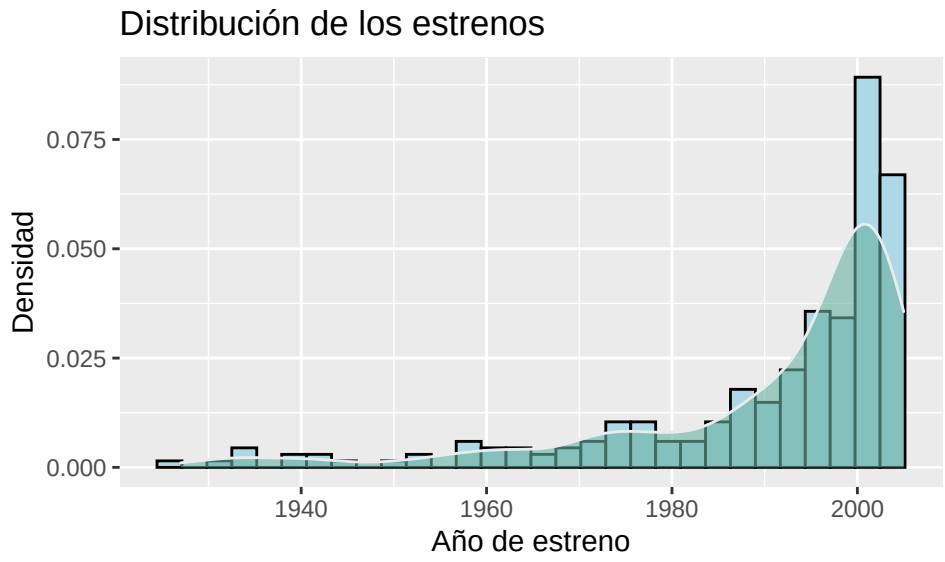


En cuanto a las valoraciones por días de la semana observamos una diferencia muy significativa entre la actividad de los primeros días de la semana y finales de semana. Antes de mitad de semana, entre el lunes y el miercoles, ya se acumulan de media un 52% de las valoraciones. Si comparamos el número de valoraciones entre los días entre semana y del fin de semana, podemos ver como solo el 21% de las valoraciones son el fin de semana, cuando un 29% de los días son fin de semana. En cuanto a la puntuación media, esta es incluso más estable que en análisis por meses, también en torno al 3.5.

3.6 Distribución películas estrenadas por año

```
movies_per_year <- titles %>%
  group_by(Release_Year) %>%
  summarise(n = n())
```

```
ggplot(data = titles,aes(x=Release_Year, y=..density..)) +
  geom_histogram(colour="black", fill="lightblue") +
  geom_density(fill="#69b3a2", color="#e9ecef", alpha=0.6) +
  labs(x='Año de estreno', y='Densidad', title='Distribución de los estrenos')
```



En nuestra muestra del 1.41% del total de las películas de netflix , el año con más estrenos fue el 2000 con 25. Por otra parte en un 50% de los años se estrenaron como mucho 2 películas.

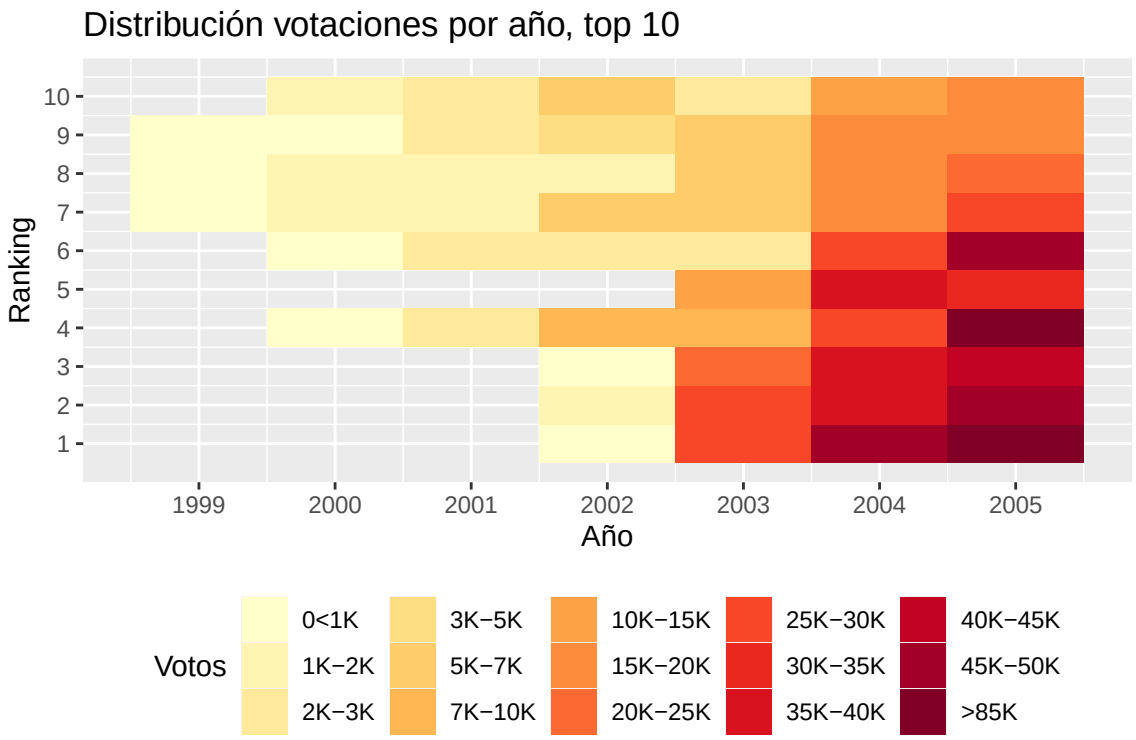
3.7 Análisis top 10 películas con más valoraciones por año de valoración

```
#agrupamos por pelicula y año en que fue valorada
votaciones_ano <- group_by(scores, MovieID, Year=year(Date)) %>%
  summarise(Votos = n_distinct((UserID)), Mean = round(mean(Score),3))
#las 10 más votadas
top10_votada <- head(arrange(movie_scores[,c('MovieID','n','Mean_Score','Title')],
                             desc(n)), 10)
movies_onfire <- filter(votaciones_ano, MovieID %in% top10_votada$MovieID)

movies_onfire$Ranking <- movies_onfire$MovieID
for (i in 1:10) {
  movie <- top10_votada$MovieID[i]
  indexes <- which(movies_onfire$MovieID == movie)
  movies_onfire$Ranking <- replace(movies_onfire$Ranking, indexes, i)
  count=i
} #Ordenamos las pelis según el top10
top10_votada$Ranking=1:10
```

Para visualizar la distribución de votaciones por año que obtubieron las 10 peliculas más votadas de *Netflix*, creamos un *Heatmap*

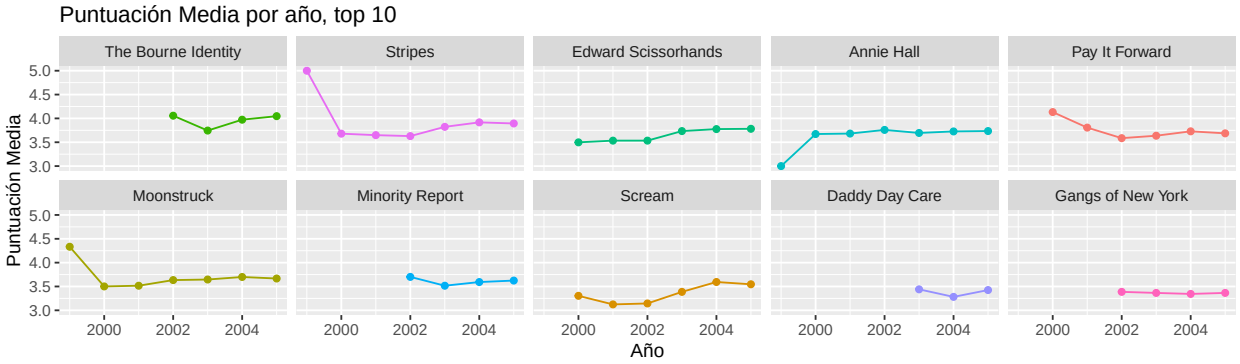
```
#movies_onfire <- arrange(movies_onfire, desc(Ranking))
#text para visualizar votos en el interactivo
ggplot(movies_onfire, aes(text = paste('Votos:', Votos), ID = MovieID,
                          y = Ranking, x = Year )) +
  geom_tile(aes(fill = secuencia)) +
  scale_y_continuous(breaks=10:1) +
  scale_x_continuous(breaks=1999:2005) +
```

```
  scale_fill_manual(values = mycolors) +              #secuencia de colores
  labs(fill = 'Votos', x='Año', title='Distribución votaciones por año, top 10') +
  theme(legend.position="bottom")
```



3.8 Evolución del score promedio de las 10 películas con más valoraciones

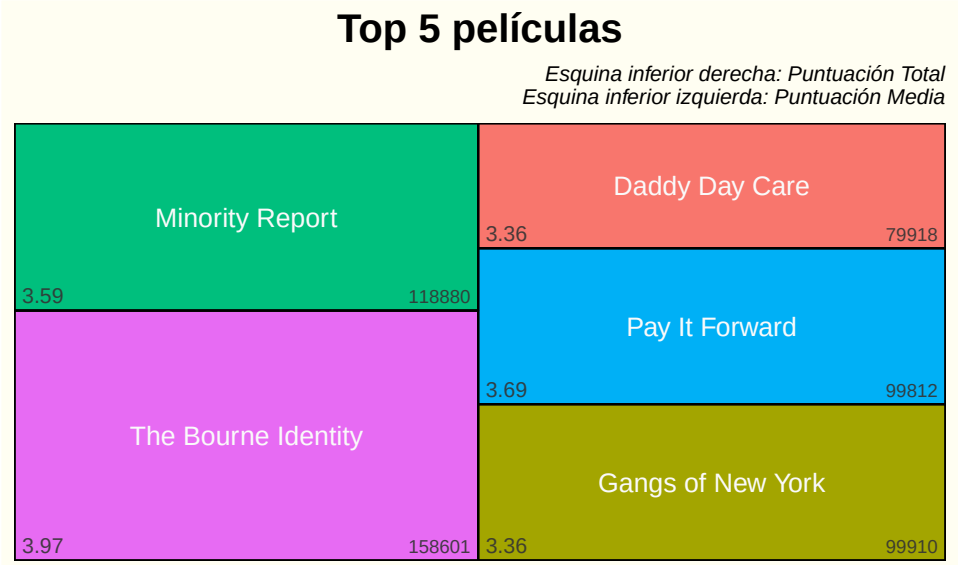
```
movies_onfire %<>% left_join(titles[,-2], by = 'MovieID')
orden_titulos <- arrange(top10_votada[,c('Title', 'Mean_Score')], desc(Mean_Score))
movies_onfire %<>% transform(Title=factor(Title, levels=as.vector(orden_titulos$Title)))

ggplot(movies_onfire, aes(Year, Mean, group=MovieID, colour=factor(MovieID))) +
  geom_point() +
  geom_line() +
  facet_wrap(~Title, nrow = 2, scale='fixed') +
  labs(y='Puntuación Media', x='Año', title='Puntuación Media por año, top 10') +
  theme(legend.position="none")
```



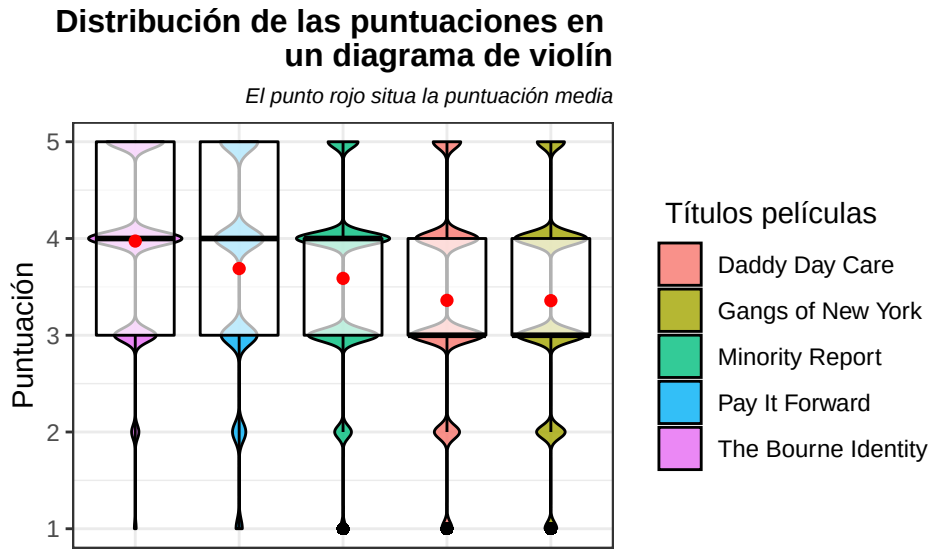
El 2000 parece que fue un punto de inflexión, 3 películas se estrenaron ese año y las ya existentes sufrieron un cambio en su puntuación media ya sea para mal como fue el caso de *Moonstruck* y *Stripes*, o para bien como en el caso de Annie Hall.

3.9 Comparación top 5 películas con más valoraciones

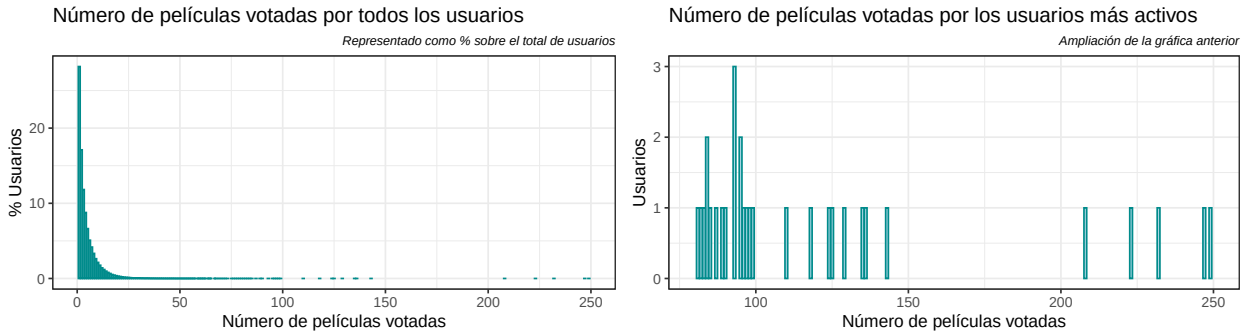
A continuación, representamos las 5 películas que recogen más valoraciones en un `treemap`:



Utilizaremos el *diagrama de violin* combinado con un diagrama de cajas y bigotes para representar la distribución de la puntuación en las 5 películas seleccionadas. El punto rojo representa la media de cada película, mientras que la línea negra que atraviesa el diagrama de cajas es la moda.



3.10 Estudios adicionales



Como se observa en los gráficos anteriores, la mayoría de usuarios no se muestran muy activos puntuando las películas, concretamente el 28.2 % de los usuarios han puntuado solamente 1 película.

Buscar los top 5 usuarios que más películas han puntuado. Luego, comparar con el top 1 usuario qué películas han dejado de evaluar el resto.

Primero vamos a buscar el número de total de películas que han sido evaluadas por usuario:

En segundo lugar, seleccionaremos el *top 5 usuarios* que más películas han puntuado,

Table 4: Top 5 Usuarios

UserID	NN	percent
305344	249	0.00017
387418	247	0.00016
2439493	232	0.00015
1664010	223	0.00015
2118461	208	0.00014

En tercer lugar, buscaremos qué películas han sido evaluadas por estos usuarios. Seguidamente, compararemos el total de películas evaluadas por el usuario `top_1` con el resto:

El usuario que más películas ha puntuado es el *305344*, entonces vamos a comparar las películas que este usuario con el resto de usuarios.

Aquí tenemos tres de las `18 películas` el `top_3` *no ha puntuado* pero el `top_1` si:

```
knitr::kable(head(films_3,3), digits = 5, col.names = NULL, align = "l", caption = "3 de las películas q
  kable_styling(latex_options = "hold_position", font_size = 8) %>%
  column_spec(1,width = "4cm")
```

\begin{table}[!h]

\caption{3 de las películas que no ha votado el top_3}

The Program
Jane Goodall's Wild Chimpanzees: IMAX
Federal Protection

\end{table}

Este proceso debería realizarse con el resto de usuarios, sin embargo, por cuestión de espacio no vamos a ejecutar los códigos. Finalmente, obtendríamos que, la diferencia del top 2 con el top 1 son *3 películas*, mientras que el top 4 con el top 1 se diferencia en *26 películas* y, por último, la diferencia con el último usuario asciende a *41 películas*.

Para terminar, hemos creado un correelograma de la tabla `scores_dates`, solamente con los valores numéricos. Luego, se ha incluido también un correelograma con los p-valores de las dimensiones anteriores, para saber si estas son o no son significantes. Tras ver los resultados, se detecta que no existe ninguna correlación importante entre las dimensiones. Solo destaca la correlación entre `n_month` y `semana`, la cual no aporta ninguna información, pues es lógico que a medida que incremente la semana, el mes también lo haga. El resto de correlaciones son muy débiles, solo destacariamos la relación entre `Score - Year` y `Release_Year - Score` pero al ser valores tan pequeños, no podemos concluir que existe correlación.