Práctica netflix

Sergio Garcia Puertas, Bartomeu Ramis Tarragó, David Cantero Tirado, Joan Jorquera Riera

7/1/2021

```
library(tidyverse)
```

```
filas_ID_combined_all=read_csv("../filas_ID_combined_all.txt")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_character(),
##   fila = col_double(),
##   ID = col_double(),
##   fila_final = col_double(),
##   data = col_double()
## )
```

Miramos la estructura de los datos

```
glimpse(filas_ID_combined_all)
```

```
## Rows: 17,770
## Columns: 5
## $ X1         <chr> "1:", "2:", "3:", "4:", "5:", "6:", "7:", "8:", "9:", "1...
## $ fila       <dbl> 1, 549, 695, 2708, 2851, 3992, 5012, 5106, 20017, 20113,...
## $ ID         <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1...
## $ fila_final <dbl> 548, 694, 2707, 2850, 3991, 5011, 5105, 20016, 20112, 20...
## $ data       <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,...
```

Miramos cuantas filas hay para cada valor de data

```
table(filas_ID_combined_all$data)
```

```
##
##    1    2    3    4
## 4499 4711 4157 4403
```

Escogemos una muestra de 250 elementos en base a la semilla generada por nuestros DNI

```
set.seed(23307791)
muestra_grupo=sample(1:17770,250, replace=FALSE)
head(muestra_grupo)
```

```
## [1]  5217  3157  5347  3858  3237 17638
```

Cuestión 1

Punto 1 - Contextualiza, a partir de la información de Kaggle, los datos de los que disponemos. Qué datos contiene cada uno de los ficheros y para qué pueden resultar importantes a Netflix.

Hemos obtenido 4 archivos desde Kaggle, que tienen el nombre de "combined_data_?". Por otro lado también tenemos los archivos "filas_ID_combined_all" y "movie_titles", proporcionados por el profesor. La "?" de "combined_data_?" coge los valores del 1 al 4. Estos 4 archivos tienen la misma estructura:

ID película:

customerID, valoración, fecha de la valoración

customerID, valoración, fecha de la valoración

.

.

.

Para cada película, habrá tantas filas como valoraciones tenga.

El archivo "filas_ID_combined_all" tiene los siguientes atributos en cada fila:

X1 -> id de la película seguido por ':'

fila -> nº de fila del archivo de Kaggle "combined_data_?" a la que hace referencia

ID -> id de la película

fila_final -> nº de fila en la que se encuentra la última referencia a esta película

data -> número del archivo de Kaggle en que se encuentra la película.

El archivo "movie_titles" tiene los siguientes atributos en cada fila:

id de la película, fecha de estreno, título de la película

Punto 2 - Leer cada película del archivo correspondiente y guardarlas, adecuadamente, en un mismo archivo para su futuro tratamiento.

Extraemos los datos de las 250 películas que hemos seleccionado y los dividimos en 4 tablas diferentes según el valor de data.

```
muestra_data1 <- filter(filas_ID_combined_all[muestra_grupo,], data==1, .preserve = TRUE)
muestra_data2 <- filter(filas_ID_combined_all[muestra_grupo,], data==2, .preserve = TRUE)
muestra_data3 <- filter(filas_ID_combined_all[muestra_grupo,], data==3, .preserve = TRUE)
muestra_data4 <- filter(filas_ID_combined_all[muestra_grupo,], data==4, .preserve = TRUE)

head(muestra_data1)
```

```
## # A tibble: 6 x 5
##   X1        fila    ID fila_final  data
##   <chr>    <dbl> <dbl>      <dbl> <dbl>
## 1 3157: 16356255  3157   16356968     1
## 2 3858: 20154315  3858   20161648     1
## 3 3237: 16664785  3237   16667097     1
## 4 3686: 19398084  3686   19400260     1
## 5 1750:  8788561  1750    8788675     1
## 6 2099: 10677090  2099   10677529     1
```

Para cada tabla leemos las películas en su archivo correspondiente de netflix junto a sus valoraciones y las guardamos en el fichero "muestrapelículas.txt". Leemos cada archivo por separado para no sobrecargar la memoria.

```
if (file.exists("muestrapeliculas.txt")) {
  file.remove("muestrapeliculas.txt")
}
```

```
## [1] TRUE
```

```
# Leemos el primer archivo de Netflix
fileName <- "../combined_data_1.txt"
archivo <- file(fileName,open="r")
datos1 <-readLines(archivo)

# Para cada fila de muestra_data1 cogemos la información indicada de "combined_data_1.txt"
# y la escribimos en "muestrapelículas.txt"
for(i in 1:nrow(muestra_data1)){
  aux <- datos1[seq(as.numeric(muestra_data1[i,2]),as.numeric(muestra_data1[i,4]))]
  write(aux,"muestrapeliculas.txt", append=TRUE)
}

# Cerramos el archivo y liberamos la memoria
remove(datos1)
close(archivo)

# Repetimos el proceso para el resto de archivos
fileName <- "../combined_data_2.txt"
archivo <- file(fileName,open="r")
datos2 <-readLines(archivo)

for(i in 1:nrow(muestra_data2)){
  aux <- datos2[seq(as.numeric(muestra_data2[i,2]),as.numeric(muestra_data2[i,4]))]
  write(aux,"muestrapeliculas.txt", append=TRUE)
}
remove(datos2)
close(archivo)

fileName <- "../combined_data_3.txt"
archivo <- file(fileName,open="r")
datos3 <-readLines(archivo)

for(i in 1:nrow(muestra_data3)){
  aux <- datos3[seq(as.numeric(muestra_data3[i,2]),as.numeric(muestra_data3[i,4]))]
  write(aux,"muestrapeliculas.txt", append=TRUE)
}
remove(datos3)
close(archivo)

fileName <- "../combined_data_4.txt"
archivo <- file(fileName,open="r")
datos4 <-readLines(archivo)
```

```
for(i in 1:nrow(muestra_data4)){
  aux <- datos4[seq(as.numeric(muestra_data4[i,2]),as.numeric(muestra_data4[i,4]))]
  write(aux,"muestrapeliculas.txt", append=TRUE)
}
remove(datos4)
close(archivo)
```

Punto 3 - Construir el modelo de datos siguiendo las instrucciones del taller de ejemplo de netflix y generar la tibble netflix

Leemos la información almacenada en "muestrapeliculas.txt".

```
fileName <- "muestrapeliculas.txt"
netflix = read_tsv(fileName, col_names = FALSE)
```

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

```
# Añadimos una columna con el nº de fila
netflix=netflix%>% mutate(fila=row_number())

# Cogemos de la tabla netflix las filas que contienen un ID de pelicula
# y las añadimos como una nueva columna
filas=grep(":",netflix$X1)
filas_ID= netflix %>%
  filter( fila %in% filas ) %>%
  mutate(ID=as.integer(gsub(":","",X1)))

# Guardamos cuantas valoraciones tiene cada pelicula
reps=diff(c(filas_ID$fila,max(netflix$fila)+1))

# Asignamos a cada valoracion la id de su pelicula correspondiente en la
# columna ID_film, eliminamos las filas que contienen unicamente un ID y
# separamos los datos de las valoraciones en las columnas ID_user, Score y date
netflix=netflix %>%
  mutate(ID1=rep(filas_ID$X1,times=reps)) %>%
  filter(!(fila %in% filas)) %>%
  select(-fila) %>%
  separate(X1,into=c("ID_user","Score","date"),sep=",") %>%
  mutate(Score=as.integer(Score)) %>%
  separate(col = ID1,into=c("ID_film","borrar")) %>%
  select(-borrar) %>% mutate(ID_film=as.numeric(ID_film))
```

Ahora la tabla netflix tiene un formato adecuado para su tratamiento.

```
head(netflix)
```

```
## # A tibble: 6 x 4
```

```
##   ID_user Score date       ID_film
##   <chr>   <int> <chr>        <dbl>
## 1 2040014     4 2004-05-19    3157
## 2 313627      1 2005-02-04    3157
## 3 115498      4 2002-02-04    3157
## 4 2251405     1 2004-09-10    3157
## 5 617861      4 2003-09-09    3157
## 6 801523      5 2005-06-17    3157
```

Punto 4 - Leer el fichero "movie_titles.csv" y hacer un inner_join para disponer del título y año de estreno de cada película.

Leemos el archivo utilizando expresiones regulares para poder leer títulos de películas que contengan comas.

```
lines   <- readLines("../movie_titles.csv")
pattern <- "^(\\d+),([^,]+),(.*)$"
matches <- regexec(pattern, lines)

bad.rows <- which(sapply(matches, length) == 1L)
if (length(bad.rows) > 0L) stop(paste("bad row: ", lines[bad.rows]))

data <- regmatches(lines, matches)
film_names <- as.data.frame(matrix(unlist(data), ncol = 4L, byrow = TRUE)[, -1L])
```

Cambiamos el tipo de dato de los atributos.

```
colnames(film_names) <- c("ID_film", "year", "title")
film_names$ID_film <- as.integer(film_names$ID_film)
film_names$year <- as.integer(film_names$year)
```

```
## Warning: NAs introducidos por coerción
```

```
summary(film_names)
```

```
##     ID_film           year         title
##  Min.   :    1   Min.   :1896   Length:17770
##  1st Qu.: 4443   1st Qu.:1985   Class :character
##  Median : 8886   Median :1997   Mode  :character
##  Mean   : 8886   Mean   :1990
##  3rd Qu.:13328   3rd Qu.:2002
##  Max.   :17770   Max.   :2005
##                  NA's   :7
```

Combinamos con un inner_join los datos de netflix y film_names para añadir los títulos de las películas y su año de estreno.

```
netflix_f <- inner_join(netflix, film_names, by="ID_film")
summary(netflix_f)
```

```
##    ID_user              Score           date              ID_film
```

```
##  Length:1146997     Min.   :1.000   Length:1146997     Min.   :   27
##  Class :character   1st Qu.:3.000   Class :character   1st Qu.: 6099
##  Mode  :character   Median :4.000   Mode  :character   Median :10730
##                     Mean   :3.646                      Mean   :10188
##                     3rd Qu.:5.000                      3rd Qu.:13347
##                     Max.   :5.000                      Max.   :17702
##       year         title
##  Min.   :1926   Length:1146997
##  1st Qu.:1985   Class :character
##  Median :1997   Mode  :character
##  Mean   :1993
##  3rd Qu.:2001
##  Max.   :2005
```

Punto 5 - Guardar los datos procesado en un fichero csv, con el formato adecuado para utilizarlo en el siguiente apartado.

Finalmente, formateamos y almacenamos los datos ya procesados en un fichero para su posterior uso.

```
netflix_f$ID_user <- as.integer(netflix_f$ID_user)
netflix_f$date <- as.Date(netflix_f$date)
netflix_f$ID_film <- as.integer(netflix_f$ID_film)
write.csv(netflix_f, "Netflix_final.csv" )
```

Cuestión 2

Cargamos los datos que hemos preparado en el apartado anterior.

```
netflix <- read.csv("Netflix_final.csv", header = TRUE)
str(netflix,width=80,strict.width="cut")
```

```
## 'data.frame':    1146997 obs. of  7 variables:
##  $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ ID_user: int  2040014 313627 115498 2251405 617861 801523 1752054 1998117 1..
##  $ Score  : int  4 1 4 1 4 5 4 4 5 1 ...
##  $ date   : chr  "2004-05-19" "2005-02-04" "2002-02-04" "2004-09-10" ...
##  $ ID_film: int  3157 3157 3157 3157 3157 3157 3157 3157 3157 3157 ...
##  $ year   : int  1964 1964 1964 1964 1964 1964 1964 1964 1964 1964 ...
##  $ title  : chr  "I Am Cuba" "I Am Cuba" "I Am Cuba" "I Am Cuba" ...
```

Punto 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. . . .

ID_user: Ya que el número del usuario no representa un valor numérico, sino que sirve para identificar al usuario, ha de ser de tipo categórico.

Score: Al tratarse de una puntuación entera, el tipo que más se ajusta es el entero.

ID_film: Ya que el numero de la película no representa un valor numérico, sino que sirve para identificar la película. Esto sumado a que su valor se repite por cada valoración que haya recibido la película hace que le asignemos el tipo categórico.

year: Los años son valores enteros.

title: Se utiliza para hacer referencia a una película. Además, se repite por cada valoración a dicha película, por tanto, se puede transformar en una variable categorica.

date: Será de tipo Date.

```
netflix <- select(netflix, -X)

netflix$ID_user <- as.factor(netflix$ID_user)
netflix$ID_film <- as.factor(netflix$ID_film)
netflix$title <- as.factor(netflix$title)

netflix$date <- as.Date(netflix$date)

str(netflix,width=80,strict.width="cut")
```

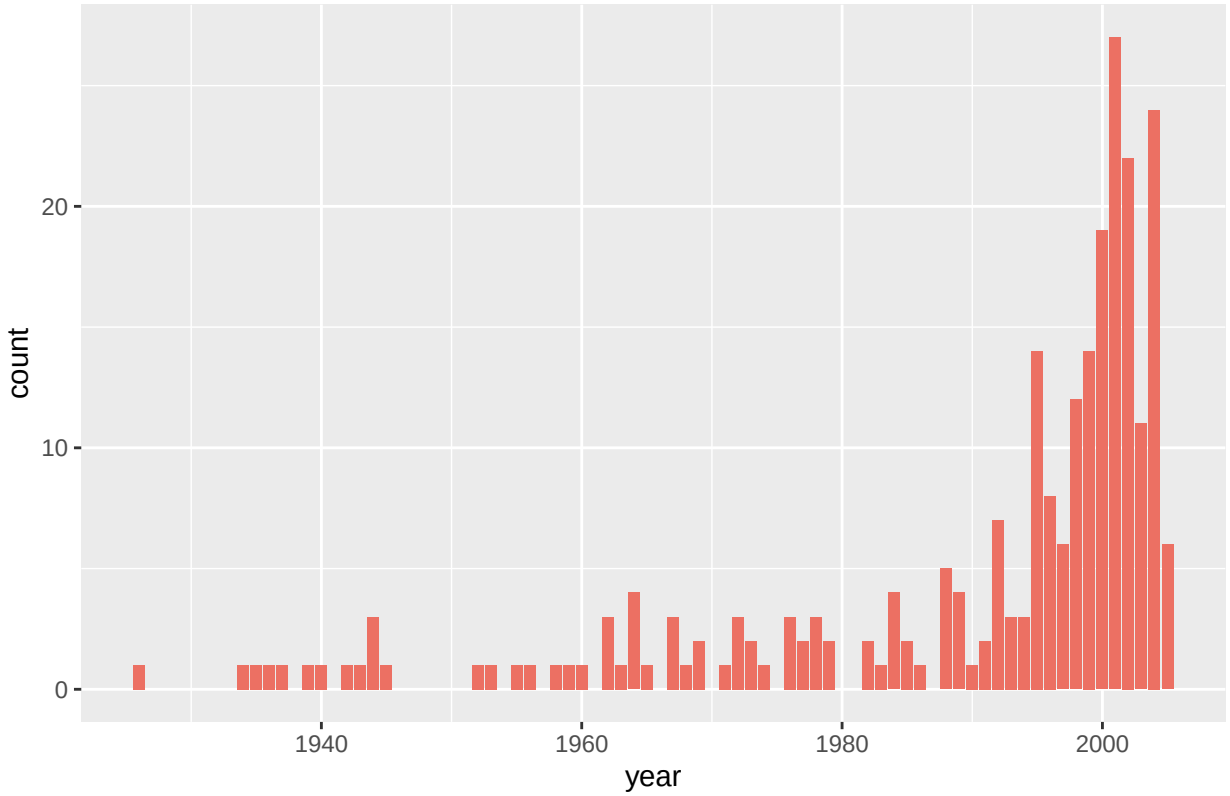
```
## 'data.frame':    1146997 obs. of  6 variables:
##  $ ID_user: Factor w/ 310843 levels "6","7","8","10",..: 239363 36800 13543 26..
##  $ Score  : int  4 1 4 1 4 5 4 4 5 1 ...
##  $ date   : Date, format: "2004-05-19" "2005-02-04" ...
##  $ ID_film: Factor w/ 250 levels "27","128","342",..: 35 35 35 35 35 35 35 35 ..
##  $ year   : int  1964 1964 1964 1964 1964 1964 1964 1964 1964 1964 ...
##  $ title  : Factor w/ 250 levels "1776","A Midsummer Night's Sex Comedy",..: 9..
```

Punto 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...)

Mostramos un gráfico que muestra el número de estrenos por año.

```
select(netflix,ID_film,year) %>% distinct %>%
ggplot + stat_count(mapping = aes(x = year), fill="#EC7063") +
  ggtitle("Fig.0: Número de películas estrenadas por año")
```

Fig.0: Número de películas estrenadas por año



En el gráfico podemos observar como Netflix parece estar más interesado en películas actuales (teniendo en cuenta que estos datos solo llegan hasta 2006) aunque también contiene un número reducido de películas clásicas.

Punto 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.

```
library(lubridate)
```

```
colnames(netflix)[5] <- "release_year"
netflix <- mutate(netflix, year = year(date),
                  month = month(date, label=TRUE),
                  week = week(date),
                  day_of_week = wday(date, week_start = 1, label = TRUE,abbr = FALSE))
str(netflix,width=80,strict.width="cut")
```

```
## 'data.frame':    1146997 obs. of  10 variables:
##  $ ID_user     : Factor w/ 310843 levels "6","7","8","10",..: 239363 36800 135..
##  $ Score       : int  4 1 4 1 4 5 4 4 5 1 ...
##  $ date        : Date, format: "2004-05-19" "2005-02-04" ...
##  $ ID_film     : Factor w/ 250 levels "27","128","342",..: 35 35 35 35 35 35 3..
##  $ release_year: int  1964 1964 1964 1964 1964 1964 1964 1964 1964 1964 ...
##  $ title       : Factor w/ 250 levels "1776","A Midsummer Night's Sex Comedy",..
```

```
##  $ year        : num  2004 2005 2002 2004 2003 ...
##  $ month       : Ord.factor w/ 12 levels "ene"<"feb"<"mar"<..: 5 2 2 9 9 6 1 4..
##  $ week        : num  20 5 5 37 36 24 5 17 52 25 ...
##  $ day_of_week : Ord.factor w/ 7 levels "lunes"<"martes"<..: 3 5 1 5 2 5 2 6 7..
```

Punto 4 - Genera una tabla que para cada película nos dé el número total de valoraciones, la suma de las valoraciones, la media de las valoraciones, y otros estadísticos de interés (desviación típica, moda , mediana).

Calculamos para cada película su numero total de valoraciones, suma de puntuación y su valoración media. También calculamos la desviación típica de las valoraciones, la moda y la mediana. Toda esta información la guardamos en la tabla estadísticos.

```
#Definimos una funcion para calcular la moda
getmode <- function(v) {
             uniqv <- unique(v)
             uniqv[which.max(tabulate(match(v, uniqv)))]
}

estadisticos <- tally(group_by(netflix,ID_film))
colnames(estadisticos) <- c("ID_film", "count")
#La función aggregate se usa para calcular las nuevas columnas, agrupando por ID de film.
estadisticos <- mutate(estadisticos,
        sum=aggregate(netflix$Score, by=list(Category=netflix$ID_film), FUN=sum)$x,
        mean=sum/count,
        sd=aggregate(netflix$Score, by=list(Category=netflix$ID_film), FUN=sd)$x,
        median=aggregate(netflix$Score, by=list(Category=netflix$ID_film), FUN=median)$x,
        mode=aggregate(netflix$Score, by=list(Category=netflix$ID_film), FUN=getmode)$x
        )

head(estadisticos)
```

```
## # A tibble: 6 x 7
##   ID_film count   sum  mean    sd median  mode
##   <fct>   <int> <int> <dbl> <dbl>  <dbl> <int>
## 1 27        273   963  3.53 1.22       4     4
## 2 128       417  1341  3.22 1.30       3     4
## 3 342      1358  5029  3.70 1.20       4     4
## 4 395      1822  7266  3.99 0.864      4     4
## 5 650       439  1175  2.68 1.20       3     3
## 6 697       209   687  3.29 1.30       4     4
```

Punto 5 - De las cinco películas con más número total de valoraciones, compara sus estadísticos y distribuciones (histogramas, boxplot, violin plot,...)

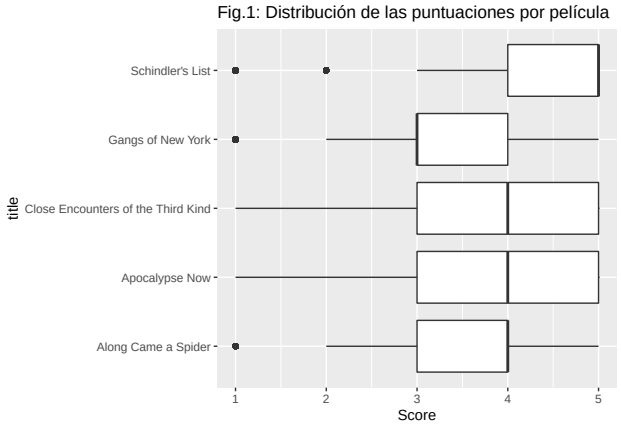
Primero, obtendremos la lista con los films más valorados, sus estadísticos y sus títulos.

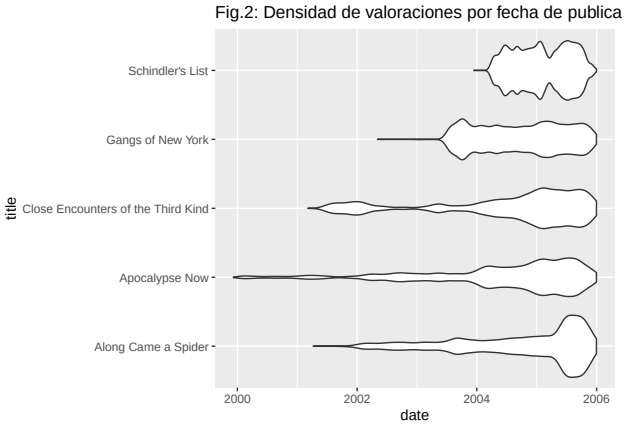
```
film_names_factor <- select(film_names,ID_film,title)
film_names_factor$ID_film <- as.factor(film_names$ID_film)

masvaloradas <- estadisticos %>% arrange(desc(count)) %>% head(5)
masvaloradas <- inner_join(masvaloradas, film_names_factor, "ID_film")
```

```
masval_netflix <- netflix %>% filter(ID_film %in% masvaloradas$ID_film)
masvaloradas
```

```
## # A tibble: 5 x 8
##   ID_film  count    sum  mean    sd median  mode title
##   <fct>    <int>  <int> <dbl> <dbl>  <dbl> <int> <chr>
## 1 12299   104800 382168  3.65 0.998      4     4 Along Came a Spider
## 2 12870   101141 450887  4.46 0.782      5     5 Schindler's List
## 3 10730    99910 335467  3.36 1.05       3     4 Gangs of New York
## 4 6099     84590 336934  3.98 1.02       4     5 Apocalypse Now
## 5 15702    51915 202498  3.90 0.895      4     4 Close Encounters of the Third ~
```

```
ggplot(data=masval_netflix, mapping = aes(x = title, y = Score)) +  geom_boxplot() +
  coord_flip() + ggtitle("Fig.1: Distribución de las puntuaciones por película")
ggplot(data=masval_netflix, mapping = aes(x = title, y = date)) + geom_violin() +
  coord_flip() + ggtitle("Fig.2: Densidad de valoraciones por fecha de publicación")
```





La figura 1 muestra la distribución de las notas. Podemos observar como las películas más populares tienen, por lo general, notas altas (entre 3 y 5). Como es lógico, estas peliculas no se han hecho populares por ser malas.

La figura 2 muestra la distribución de las fechas en las que se hicieron las valoraciones. Aquí vemos como pasa cierto tiempo entre que una película es añadida al catálogo y entre que se populariza. Además, también podemos suponer que, alrededor del 2004 Netflix tuvo un auge de clientes nuevos.

```
ggplot(data=masvaloradas, mapping = aes(x = title, y=sd)) + geom_col(fill="#EC7063") +
  ggtitle("Fig.3: Comparación de la esviación estándar") +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=1))
ggplot(data=masval_netflix, mapping = aes(x = date)) + geom_histogram(fill="#EC7063") +
  ggtitle("Fig.4: Valoraciones totales por fecha de publicación")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

Fig.3: Comparación de la esviación estándar

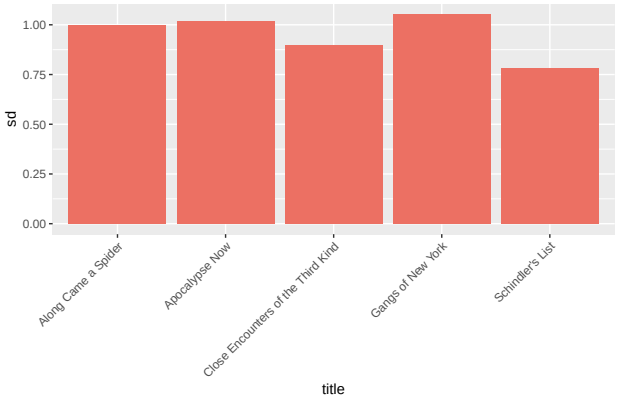
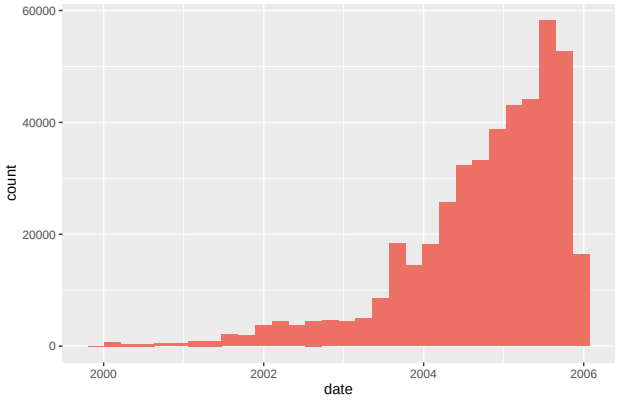


Fig.4: Valoraciones totales por fecha de publicación



En la figura 3, comparamos la desviación estándar de las 5 películas más valoradas. Como vemos, esta desviación es bastante similar. El film donde las notas de las valoraciones són más homogenias, es decir, la desviación es menor, es "Schindler's List", con una desviación típica ligeramente superior a 0,75.

Por último, vemos claramente en la figura 4, que se confirma la hipótesis de la figura 2: Netflix sufrió un auge de usuarios, y por tanto de valoraciones, alrededor del 2004.

Punto 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?

```
ggplot(data=netflix, mapping = aes(x = month)) +
  geom_bar(fill="#EC7063") +
  ggtitle("Fig.5: Número de valoraciones por meses")
ggplot(data=netflix, mapping = aes(x = day_of_week)) +
  geom_bar(fill="#EC7063") +
  ggtitle("Fig.6: Número de valoraciones por días de la semana")
```

Fig.5: Número de valoraciones por meses

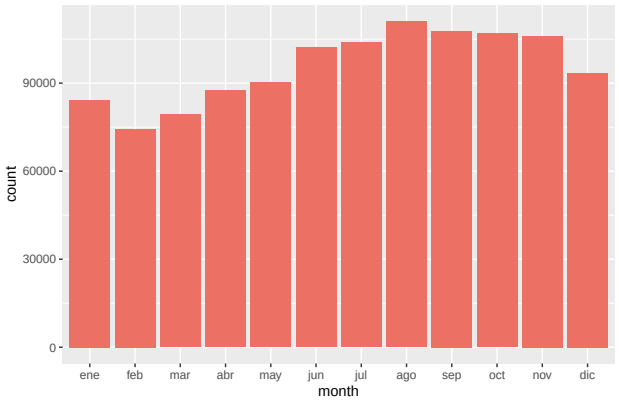
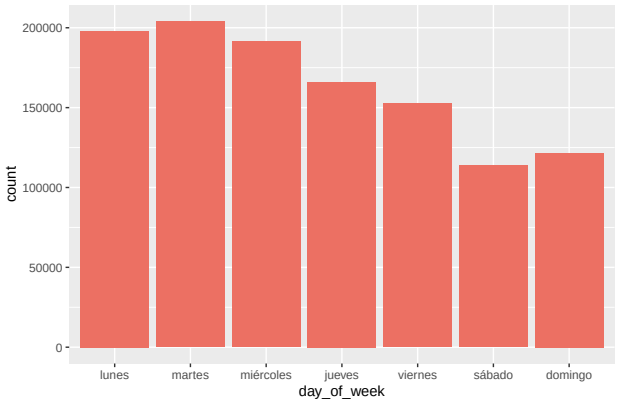


Fig.6: Número de valoraciones por días de la semana

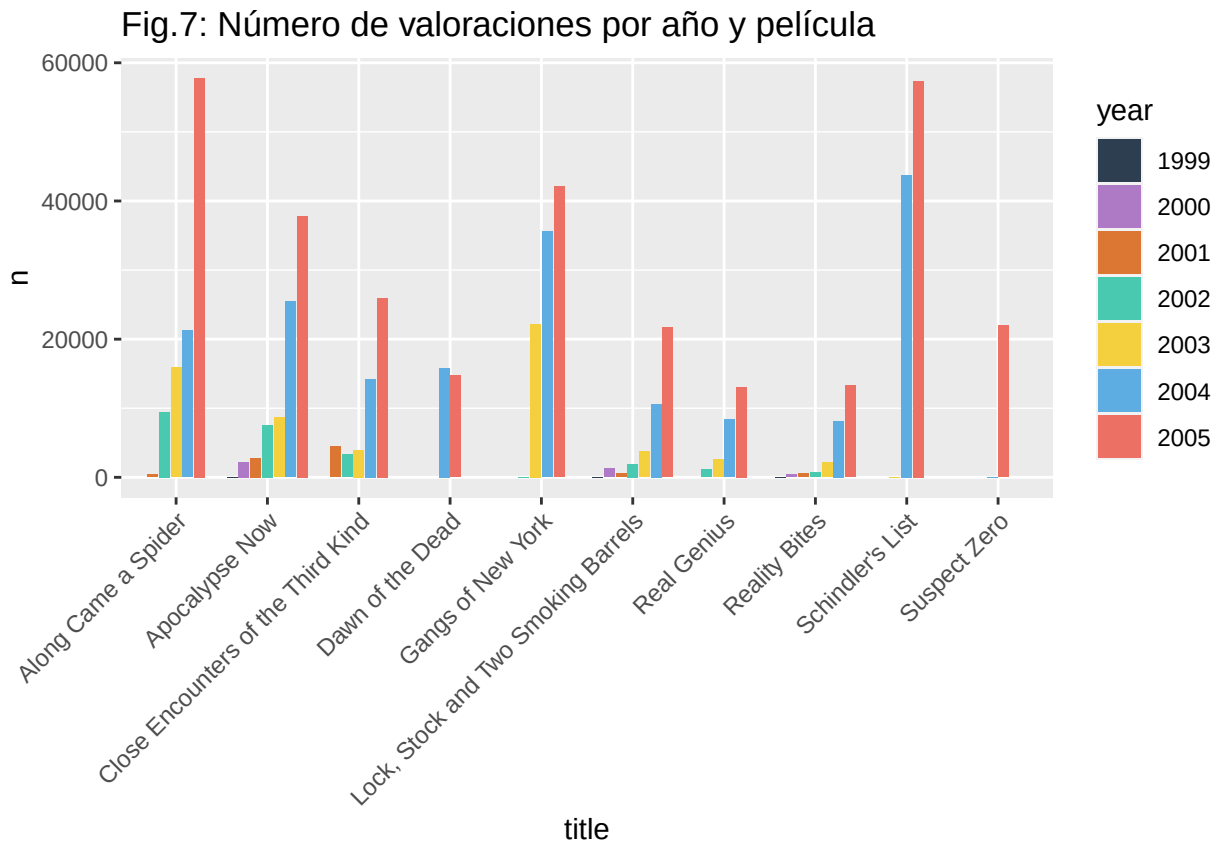


Suponemos, viendo la figura 6, que en los fines de semana cuando la gente tiene más tiempo libre, realizan otras actividades (probablemente más sociales fuera de casa) en lugar de ver netflix. Por tanto, los dias entre semana al estar en casa aprovechan para ver películas.

Como la mayoría de las valoraciones se han hecho en un plazo de 2 años, no podemos sacar conclusiones claras, a partir de la figura 5, sobre la distribución de las valoraciones según los meses.

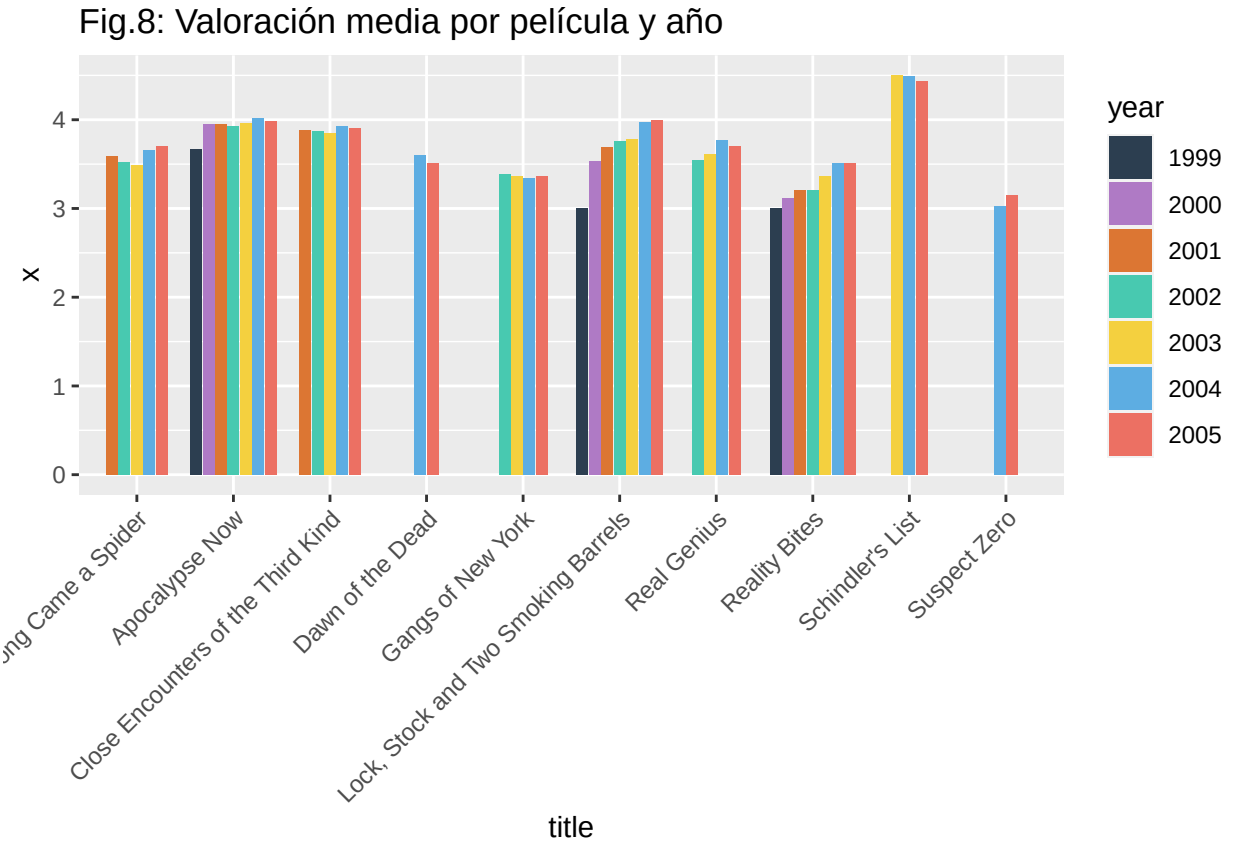
Punto 7 - Genera una tabla agrupada por película y año del número de valoraciones. Representa la tabla gráficamente para las 10 películas con mayor número de valoraciones.

```
#Obtenemos para cada pelicula y cada año, cuantas valoraciones se han hecho
valoracionXaño <- netflix %>% group_by(ID_film,year) %>% tally

#Añadimos a la tabla el título de cada película
valoracionXaño <- valoracionXaño %>% inner_join(film_names_factor,by="ID_film")
#Dejamos solo los datos de las 10 películas con más valoraciones
diezmasvaloradas <- estadisticos %>% arrange(desc(count)) %>% head(10)
aux <- valoracionXaño %>%
  filter(ID_film %in% diezmasvaloradas$ID_film) %>%
  arrange(desc(n))
aux$year <- as.factor(aux$year)
#Dibujamos la gráfica con la información obtenida
ggplot(data=aux, mapping = aes(x = title, y=n, fill=year)) +
  geom_bar(stat="identity", position=position_dodge2(preserve = "single")) +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=1)) +
  scale_fill_manual(values=c("#2C3E50","#AF7AC5","#DC7633",
                             "#48C9B0","#F4D03F","#5DADE2","#EC7063")) +
  ggtitle("Fig.7: Número de valoraciones por año y película")
```



Punto 8 - Distribución del score promedio por año de las 10 películas con mayor número de valoraciones.

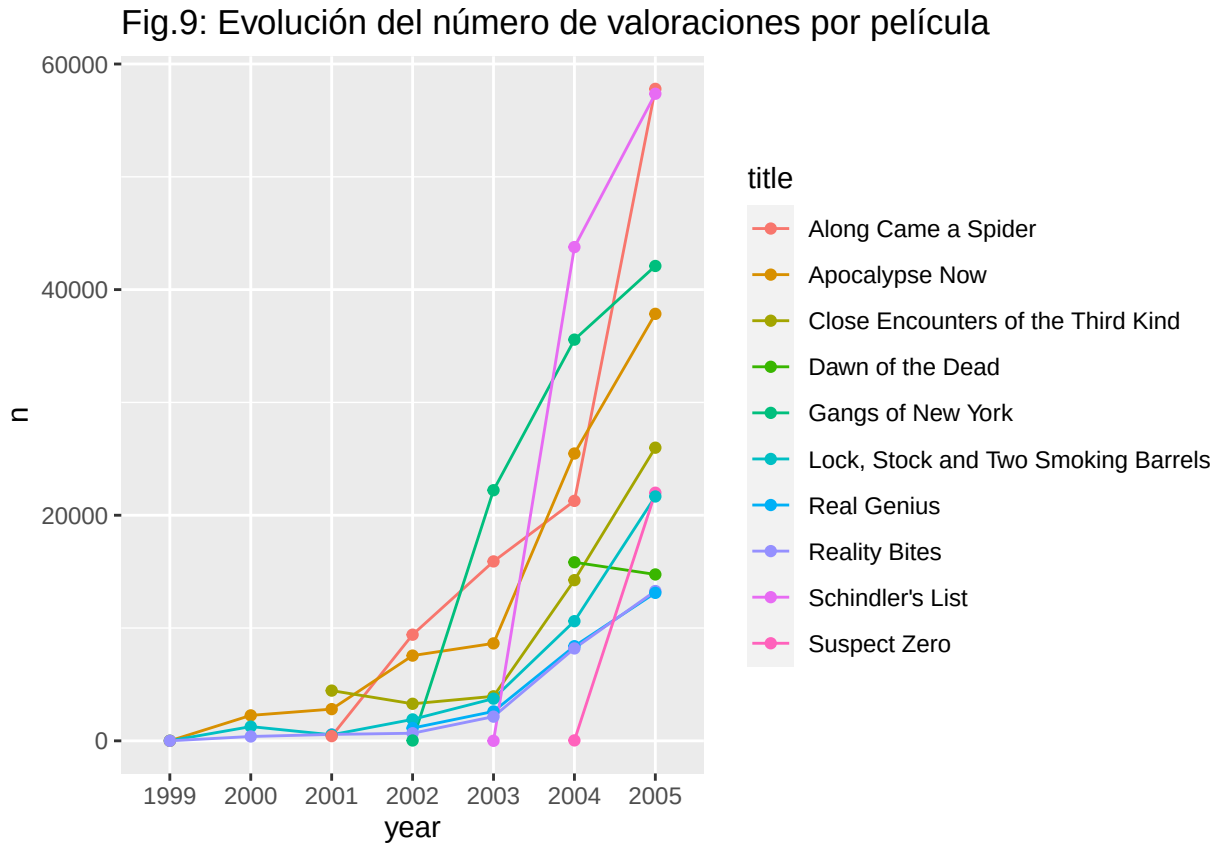
```
#Calculamos la media de las valoraciones para cada película y cada año
aux2 <- aggregate(netflix$Score, by=list(title=netflix$title,ID_film=netflix$ID_film,
                                         year=netflix$year), FUN=mean)
#Nos quedamos solo con las 10 películas más valoradas
aux2 <- aux2 %>%
  filter(ID_film %in% diezmasvaloradas$ID_film) %>%
  arrange(desc(x))
aux2$year <- as.factor(aux2$year)

ggplot(data=aux2, mapping = aes(x = title, y=x, fill=year)) +
  geom_bar(stat="identity", position=position_dodge2(preserve = "single")) +
  theme(axis.text.x=element_text(angle=45,hjust=1,vjust=1)) +
  scale_fill_manual(values=c("#2C3E50","#AF7AC5","#DC7633",
                             "#48C9B0","#F4D03F","#5DADE2","#EC7063")) +
  ggtitle("Fig.8: Valoración media por película y año")
```



Como podemos observar, las notas que se dan a una película no varían demasiado en función del año.

Punto 9 - Realiza algún gráfico o estudio estadístico adicional que consideres informativo en base al análisis exploratorio anterior

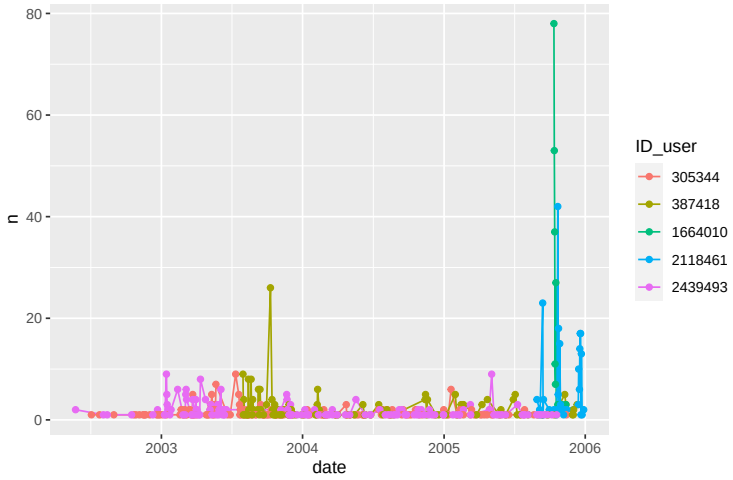
```
ggplot(data=aux, aes(x=year, y=n, group=title)) +
  geom_line(aes(color=title))+
  geom_point(aes(color=title)) +
  ggtitle("Fig.9: Evolución del número de valoraciones por película")
```



A continuación vamos a mirar cuantas valoraciones hacen los usuarios en cada fecha. En concreto nos centraremos en los 5 usuarios con más valoraciones totales.

```
usuarios <- tally(group_by(netflix,ID_user))
usuarios <- arrange(usuarios,desc(n))
usuarios <- head(usuarios,5)

usuariosFechas <- tally(group_by(netflix,ID_user,date))
usuariosFechas <- arrange(usuariosFechas,desc(n))

usuariosFechas <-subset(usuariosFechas, usuariosFechas$ID_user %in% usuarios$ID_user )

ggplot(data=usuariosFechas, aes(x=date, y=n, group=ID_user)) +
  geom_line(aes(color=ID_user)) +
  geom_point(aes(color=ID_user)) +
  ggtitle("Fig.10: Valoraciones de los usuarios por fecha")
```

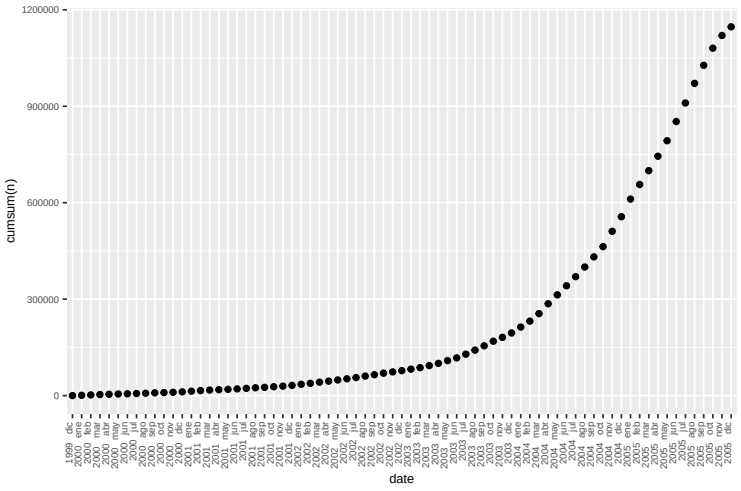
Fig.10: Valoraciones de los usuarios por fecha



A partir de los datos que tenemos no se puede realizar un estudio basado en las fechas en las que se ven las películas. Como podemos observar en el gráfico anterior, muchas de las valoraciones no se realizan el mismo dia en que el usuario ve la película ya que probablemente una persona no haya visto 78 peliculas el mismo día.

```
sumaValoraciones <- tally(group_by(netflix,year, month))
sumaValoraciones$date <- paste(sumaValoraciones$year," ",sumaValoraciones$month)
sumaValoraciones$date <- as.factor(sumaValoraciones$date)

ggplot(sumaValoraciones, aes(x=date, y=cumsum(n))) + geom_point()+
  scale_x_discrete(limits=sumaValoraciones$date) +
  theme(axis.text.x=element_text(angle=90,hjust=1,vjust=0),text = element_text(size=8)) +
  ggtitle("Fig.11: Evolución del número total de valoraciones")
```

Fig.11: Evolución del número total de valoraciones



Aqui podemos ver la evolución de la popularidad y uso de netflix. A partir de 2004 aumenta el ratio de crecimiento de la cantidad de valoraciones que se hacen al mes indicando una mayor cantidad de clientes.

A continuación intentaremos realizar un clustering en el que agrupemos los usuarios en base a las películas que han visto con tal de poder hacer recomendaciones personalizadas. Para la implementación de un sistema

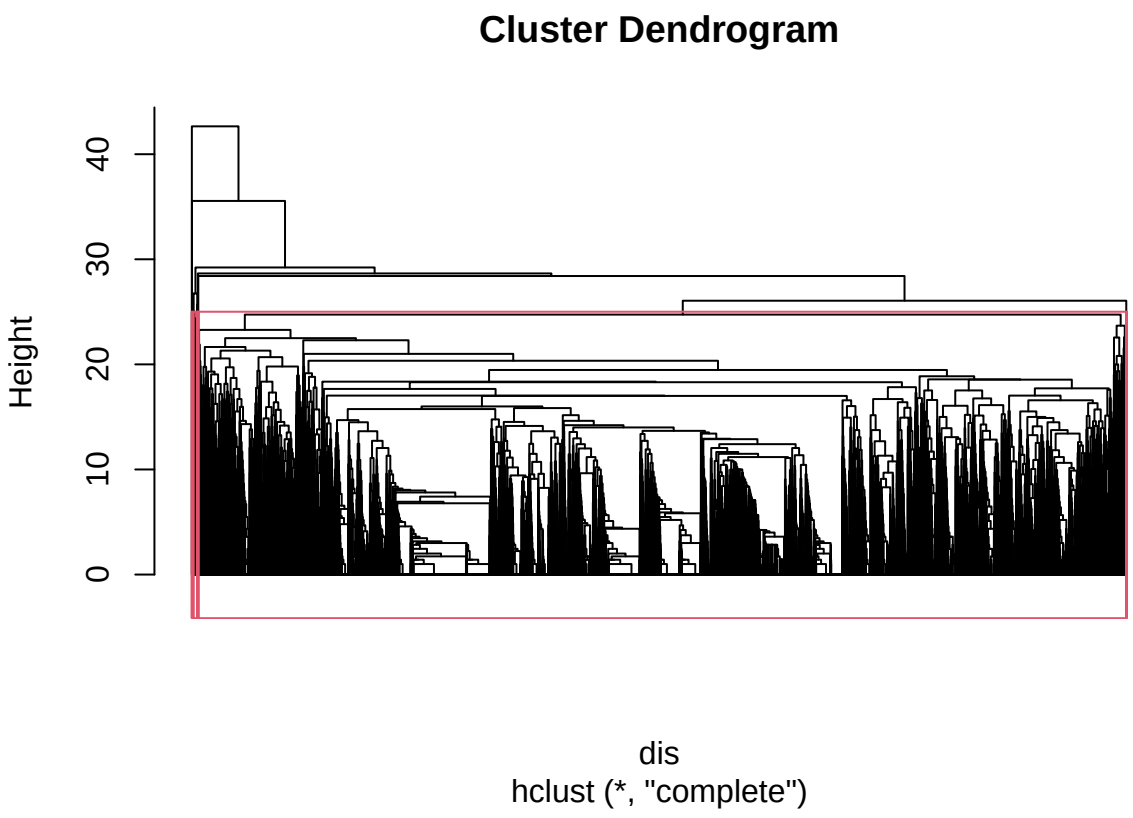
de recomendación sería más óptimo agrupar las películas en función de sus temáticas. Así, si un usuario da una buena valoración a una película, se le podrían recomendar más películas del mismo cluster. En nuestro caso, no disponemos de datos que nos permitan evaluar la similitud entre películas, por lo que utilizaremos un enfoque distinto. Sabiendo qué valoración ha dado cada usuario a cada película podemos hacer comparaciones entre usuarios y de esta forma calcular una "similitud" entre usuarios. El sistema de recomendación funcionaria de manera que a un usuario se le recomendarían películas a las que otro usuario de su mismo cluster ha dado una buena valoración.

```
#Obtener tabla con el formato correcto (Id_user, puntuacion_peli1, puntuacion_peli2, ...)
model.df <- netflix %>% select(c(ID_user, title, Score)) %>% spread( title, Score)
```

En model.df tenemos una fila por usuario con la valoración que le ha dado a cada película. Si alguna no la ha valorado, aparece como NA.

```
#Generar matriz de distancias entre usuarios
#No hace falta normalizar los datos porque todos los atributos son notas de películas)
muestra = sample(1:310843,10000, replace=FALSE)
model.reduced <- model.df[muestra,]
model.reduced <- select(model.reduced, -ID_user)
model.reduced[is.na(model.reduced)]=0
dis <- dist(model.reduced , method="euclidean")
```

```
hc <- hclust(dis, method = "complete")
plot(hc, labels=FALSE, hang=-1)
rect.hclust(hc, k=10)
groups <- cutree(hc, k=10)
```



Tras este proceso, obtenemos un modelo de clustering jerárquico con 10 clusters. Cada uno de estos define un "perfil de usuario", es decir, cada conjunto representan un grupo de clientes con gustos "similares". Como ya hemos explicado anteriormente, las fechas resultan poco útiles, pues no representan el dia real en que cada usuario vio la película. Esto por si solo nos obliga a basarnos completa y exclusivamente en las valoraciones. Otro aspecto a considerar, es la escasez de puntuaciones de cada usuario. La gran mayoría de estos solo han valorado entre 1 y 2 películas, y por tanto, resulta muy complejo determinar de una forma minímamente concreta sus gustos. Es por esto, que al usar las técnicas de clustering, muchos de estas agrupaciones de clientes están formadas por un solo usuario. Además, por problemas computacionales, al haber necesitado reducir la muestra de clientes, se nos ha limitado aún más a la hora de poder analizar estos datos.

Tras probar varios tipos de clustering cambiando la distancia utilizada y probando distintos métodos de linkage, hemos decidido dejar en el documento la versión que nos ha dado mejores resultados. Hemos decidido agrupar a los usuarios en 10 clusters, pero por los datos que tenemos para resolver el problema, los grupos que se diferencian no son útiles ya que hay un grupo inmenso, con prácticamente todos los usuarios, y 9 grupos muy pequeños. Por tanto estos clusters no recomendarían películas adecuadamente.