Nombre: Celic Gabriel Hernández Archundia

Matrícula: 2877240

```
%%capture
!pip·install·rpy2==3.5.1
```

```
%load_ext rpy2.ipython
```

```
from google.colab import drive
drive.mount('/content/drive',force_remount=True)
```

```
Mounted at /content/drive
```

```
%%R
library(readr)
library(magrittr)
library(dplyr)
```

```
WARNING:rpy2.rinterface_lib.callbacks:R[write to console]:
Attaching package: ‘dplyr’


WARNING:rpy2.rinterface_lib.callbacks:R[write to console]: The following objects are masked from ‘package:stats’:

    filter, lag


WARNING:rpy2.rinterface_lib.callbacks:R[write to console]: The following objects are masked from ‘package:base’:

    intersect, setdiff, setequal, union
```

```
%%R
datos <- read.csv("drive/MyDrive/Tecmilenio/Big Data/movies.csv")
```

0. ¿Cuántas películas tiene el conjunto de datos?

```
%%R
length(rownames(datos))
```

```
[1] 7668
```

1. ¿Cuáles son los diferentes tipos de clasificación (Rating) y cuántos registros existen por cada uno?

```
%%R
colnames(datos)
```

```
 [1] "name"     "rating"   "genre"    "year"     "released" "score"
 [7] "votes"    "director" "writer"   "star"     "country"  "budget"
[13] "gross"    "company"  "runtime"
```

```
%%R
datos %>%
group_by(rating) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
summarize(cantidad = n()) # n() -> para contar datos
```

```
# A tibble: 13 × 2
   rating      cantidad
   <chr>          <int>
 1 ""                77
 2 "Approved"         1
 3 "G"              153
 4 "NC-17"           23
 5 "Not Rated"      283
 6 "PG"            1252
```

```
 7 "PG-13"         2112
 8 "R"             3697
 9 "TV-14"            1
10 "TV-MA"            9
11 "TV-PG"            5
12 "Unrated"         52
13 "X"                3
```

2. ¿Cuáles son los diferentes tipos de género (Genre) y cuántos registros existen por cada uno?

```
%%R
datos %>%
group_by(genre) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
summarize(cantidad = n()) # n() -> para contar datos
```

```
# A tibble: 19 × 2
   genre     cantidad
   <chr>        <int>
 1 Action        1705
 2 Adventure      427
 3 Animation      338
 4 Biography      443
 5 Comedy        2245
 6 Crime          551
 7 Drama         1518
 8 Family          11
 9 Fantasy         44
10 History          1
11 Horror         322
12 Music            1
13 Musical          2
14 Mystery         20
15 Romance         10
16 Sci-Fi          10
17 Sport            1
18 Thriller        16
19 Western          3
```

3. ¿Cuántas películas hay registradas por cada año?

```
%%R
datos %>%
group_by(year) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
summarize(cantidad = n()) %>% # n() -> para contar datos
print(n=41)
```

```
# A tibble: 41 × 2
    year cantidad
   <int>    <int>
 1  1980       92
 2  1981      113
 3  1982      126
 4  1983      144
 5  1984      168
 6  1985      200
 7  1986      200
 8  1987      200
 9  1988      200
10  1989      200
11  1990      200
12  1991      200
13  1992      200
14  1993      200
15  1994      200
16  1995      200
17  1996      200
18  1997      200
19  1998      200
20  1999      200
21  2000      200
22  2001      200
23  2002      200
24  2003      200
25  2004      200
26  2005      200
27  2006      200
28  2007      200
29  2008      200
```

```
30  2009      200
31  2010      200
32  2011      200
33  2012      200
34  2013      200
35  2014      200
36  2015      200
37  2016      200
38  2017      200
39  2018      200
40  2019      200
41  2020       25
```

4. En promedio, ¿Qué año tiene el Score más alto?

```
%%R
datos %>%
group_by(year) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
summarize(col = mean(score)) %>% # n() -> para contar datos
arrange(desc(col)) %>% # arrange es para acomodar en orden ascendente o descendente
print(n=41)
```

```
# A tibble: 41 × 2
    year   col
   <int> <dbl>
 1  2016  6.62
 2  2013  6.62
 3  2014  6.59
 4  2017  6.56
 5  2015  6.52
 6  2004  6.52
 7  1999  6.50
 8  2018  6.49
 9  2011  6.48
10  2007  6.47
11  2012  6.47
12  2006  6.46
13  1998  6.46
14  1995  6.46
15  2010  6.46
16  2001  6.45
17  2009  6.44
18  1992  6.41
19  1991  6.39
20  2008  6.39
21  1993  6.38
22  2005  6.36
23  2019  6.36
24  2002  6.36
25  2000  6.36
26  1997  6.36
27  2003  6.33
28  1990  6.33
29  1994  6.31
30  1985  6.31
31  1980  6.30
32  1981  6.30
33  1982  6.29
34  1988  6.28
35  1996  6.24
36  1987  6.22
37  1984  6.19
38  1989  6.18
39  1986  6.15
40  1983  6.02
41  2020 NA
```

5. ¿Qué año tiene la cantidad de votos más alta?

```
%%R
datos %>%
group_by(year) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
summarize(col = sum(votes)) %>% # n() -> para contar datos
arrange(desc(col)) %>% # arrange es para acomodar en orden ascendente o descendente
print(n=1)
```

```
# A tibble: 41 × 2
   year      col
  <int>    <dbl>
1  2013 33093300
# ℹ 40 more rows
# ℹ Use `print(n = ...)` to see more rows
```

6. ¿Qué directores han filmado más de 5 películas?

```
%%R
colnames(datos)
```

```
 [1] "name"     "rating"   "genre"    "year"     "released" "score"
 [7] "votes"    "director" "writer"   "star"     "country"  "budget"
[13] "gross"    "company"  "runtime"
```

```
%%R
datos %>%
group_by(director) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
summarize(col = n()) %>% # n() -> para contar datos
arrange(desc(col)) %>% # arrange es para acomodar en orden ascendente o descendente
print(n=5)
```

```
# A tibble: 2,949 × 2
  director           col
  <chr>            <int>
1 Woody Allen         38
2 Clint Eastwood      31
3 Directors           28
4 Steven Spielberg    27
5 Ron Howard          24
# ℹ 2,944 more rows
# ℹ Use `print(n = ...)` to see more rows
```

7. ¿Quién es el actor protagonista que participó en más películas en cada década (80s, 90s, 00s, 10s, 20s)?

```
%%R
datos %>%
mutate(decade = case_when(
    year >= 1980 & year < 1990 ~ "80s",
    year >= 1990 & year < 2000 ~ "90s",
    year >= 2000 & year < 2010 ~ "00s",
    year >= 2010 & year < 2020 ~ "10s",
    year >= 2020 & year < 2030 ~ "20s"
)) %>%
select(decade,star) %>%
group_by(decade, star) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
summarize(num_peliculas = n()) %>% # n() -> para contar datos
summarize(mejor_star = star[which.max(num_peliculas)], num_pelis = max(num_peliculas))
```

```
`summarise()` has grouped output by 'decade'. You can override using the
`.groups` argument.
# A tibble: 5 × 3
  decade mejor_star     num_pelis
  <chr>  <chr>              <int>
1 00s    Nicolas Cage          17
2 10s    Dwayne Johnson        14
3 20s    Augie Tulba            1
4 80s    Burt Reynolds         14
5 90s    Bruce Willis          15
```

8. Top 10 de las películas con más presupuesto

```
%%R
datos %>%
select(name, budget) %>% # Tú quieres contar, promediar, etc., cuando usas groupby
arrange(desc(budget)) %>% # arrange es para acomodar en orden ascendente o descendente
head(n=10)
```

```
                                        name   budget
1                           Avengers: Endgame 3.56e+08
2                         Avengers: Infinity War 3.21e+08
```

```
3       Star Wars: Episode VIII - The Last Jedi 3.17e+08
4      Pirates of the Caribbean: at World's End 3.00e+08
5                                Justice League 3.00e+08
6                       Solo: A Star Wars Story 2.75e+08
7  Star Wars: Episode IX - The Rise of Skywalker 2.75e+08
8                               Superman Returns 2.70e+08
9                                        Tangled 2.60e+08
10                                The Lion King 2.60e+08
```

✓ 0 s se ejecutó 18:14