PRA2-TYC

Juan Miguel Iglesias Labraca, Elías Páez de la Rosa

04/01/2022

Práctica 2

Descripción del dataset. ¿Por qué es importante y qué pregunta/problema pretende responder?

El conjunto de datos muestra los **20.000** juegos de mesa mejor valorados de la web BoardGameGeek.

Dicho dataset ha sido facilitado por Dilini Samarasinghe de la Universidad de New South Wales, Australia en la siguiente ruta: *https://ieee-dataport.org/open-access/boardgamegeek-dataset-board-games*.

Los datos se extrajeron por medio de web scraping a fecha 07/05/2021.

Los campos que forman en dataset son los siguientes:

- **ID:** código identificador del juego.
- **Name:** Nombre del juego.
- **Year Published:** Año de publicación del juego.
- **Min Players:** Número mínimo de jugadores.
- **Max Players:** Número máximo de jugadores.
- **Play Time:** Tiempo estimado de cada partida.
- **Min Age:** Edad mínima recomendada para el juego.
- **Users Rated:** Cantidad de usarios que han valorado el juego.
- **Rating Average:** Media de la puntuación otorgada por los usuarios.
- **BGG Rank:** Ranking del juego respecto al resto.
- **Complexity Average:** Complejidad o dificultad del juego, valoración media realizada por los usuarios.
- **Owned Users:** Cantidad de usuarios que afirman tener el juego.
- **Mechanics:** Campo con múltiples valores, cada valor indica una mecánica del juego (lanzar dados, robar cartas, por turnos, etc.).
- **Domains:** Género del juego, campo con hasta 2 valores simultáneamente (Familiar, Estrategia, Infantil, etc.).

En el mundo de los juegos de mesa, BoardGameGeek contiene la mayor librería de datos al respecto, de ahí que hayamos optado por un dataset que provenga de dicha web.

Se pretende responder a las siguientes preguntas:

-¿Cuales son los géneros (Domains) más vendidos (Owned.Users)?

-¿Cuales son las mecánicas (Mechanics) mas comunes?

-¿Influye el género (Domains) en la valoración media del juego (Rating.Average)?

-¿Qué factores influyen más en la duración de las partidas (Play.Time)?

-¿El año de publicación influye en alguna otra variable?

-¿Qué sería necesario para hacer una estimación de las ventas?

Integración y selección de los datos de interés a analizar

Empezamos cargando el dataset en nuestro entorno de RStudio para posteriormente analizar los diferentes campos que lo componen y poder pasar más adelante a su limpieza.

```
# Configuramos el working directory y leemos el CSV
setwd("E:/Main")
datos <- read.csv("BGG_Data_Set.csv", sep=";", encoding='Latin-1')
```

```
# Mostramos los campos del dataset
str(datos)
```

```
## 'data.frame':    20343 obs. of  14 variables:
##  $ ID                : int  174430 161936 224517 167791 233078 291457 182028 220308 187645 12333 ...
##  $ Name              : chr  "Gloomhaven" "Pandemic Legacy: Season 1" "Brass: Birmingham" "Terraformi
##  $ Year.Published    : int  2017 2015 2018 2016 2017 2020 2015 2017 2016 2005 ...
##  $ Min.Players       : int  1 2 2 1 3 1 2 1 2 2 ...
##  $ Max.Players       : int  4 4 4 5 6 4 4 4 4 2 ...
##  $ Play.Time         : int  120 60 120 120 480 120 120 150 240 180 ...
##  $ Min.Age           : int  14 13 14 12 14 14 14 12 14 13 ...
##  $ Users.Rated       : int  42055 41643 19217 64864 13468 8392 23061 16352 23081 40814 ...
##  $ Rating.Average    : num  8.79 8.61 8.66 8.43 8.7 8.87 8.43 8.49 8.42 8.29 ...
##  $ BGG.Rank          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ Complexity.Average: num  3.86 2.84 3.91 3.24 4.22 3.55 4.41 4.35 3.71 3.59 ...
##  $ Owned.Users       : int  68323 65294 28785 87099 16831 21609 26985 20312 34849 56219 ...
##  $ Mechanics         : chr  "Action Queue, Action Retrieval, Campaign / Battle Card Driven, Card Play
##  $ Domains           : chr  "Strategy Games, Thematic Games" "Strategy Games, Thematic Games" "Strate
```

Seleccionamos todos los campos para trabajar posteriormente con ellos, a excepción de **BGG.Rank**, que lo eliminamos del dataset ya que no aporta valor:

```
# Eliminamos el campo BGG.Rank
datos$BGG.Rank <- NULL
```

Limpieza de los datos

¿Los datos contienen ceros o elementos vacíos? ¿Cómo gestionarías cada uno de estos casos?

Debido a que hay algunos registros corruptos, los campos **ID**, **Year.Published** y **Owned.Users** contienen valores vacíos. En total son solo **23 registros** que decidimos eliminar ya que suponen tan solo un 0,1% del total y no aportan valor.

```
colSums(is.na(datos))
```

```
##                 ID               Name     Year.Published        Min.Players
##                 16                  0                  1                  0
##        Max.Players          Play.Time            Min.Age       Users.Rated
##                  0                  0                  0                  0
##     Rating.Average Complexity.Average        Owned.Users          Mechanics
##                  0                  0                 23                  0
##            Domains
##                  0
```

```
# Eliminamos los registros
datos <- na.omit(datos)
```

Por otro lado buscamos valores no informados y encontramos que los campos **Mechanics** y **Domains** son los únicos que tienen esta casuística.

Se decide pasarles el valor "Unknown" a dichos registros para no tener que eliminarlos y para visualizarlo mejor más adelante.

```
colSums(datos == "")
```

```
##                 ID               Name     Year.Published        Min.Players
##                  0                  0                  0                  0
##        Max.Players          Play.Time            Min.Age       Users.Rated
##                  0                  0                  0                  0
##     Rating.Average Complexity.Average        Owned.Users          Mechanics
##                  0                  0                  0               1581
##            Domains
##              10136
```

```
datos$Domains[datos$Domains == ""] <- "Unknown"
datos$Mechanics[datos$Mechanics == ""] <- "Unknown"
```

Para evitar posteriores errores realizamos las siguientes transformaciones:

```
# Convertimos las variables character a factor
for (i in c(2:length(datos))){
  if(!is.numeric(datos[,i]))
    datos[,i] = as.factor(datos[,i])
}
```

Se detectan 25 juegos con carácteres corruptos, posiblemente por la transcripción de los datos en un formato UTF incompatible, por lo que se procede a eliminarlos.

```
# Contamos antes y después para confirmar
nrow(datos[grep("\\?\\?", datos$Name),])
```

```
## [1] 25
```

```
datos <- datos[- grep("\\?\\?", datos$Name),]
nrow(datos[grep("\\?\\?", datos$Name),])
```

```
## [1] 0
```

Identificación y tratamiento de valores extremos.

Buscamos outliers en el campo **Year.Published**, para ello creamos un nuevo campo **decada** en el que se guarda la década a la que pertenece el juego y posteriormente mostramos con las funciones sort() y $out los outliers de forma cronológica.

```
datos$decada <- round(datos$Year.Published/10)
sort(unique(boxplot.stats(datos$decada)$out))
```

```
##  [1] -350 -300 -260 -220 -140 -130  -20  -10    0   40   50   55   60   65   70
## [16]   76  100  112  115  130  140  142  144  148  150  153  155  159  160  163
## [31]  166  168  169  170  172  174  175  176  178  180  181  182  183  184  185
## [46]  186  187  188  189  190  191  192  193  194  195  196
```

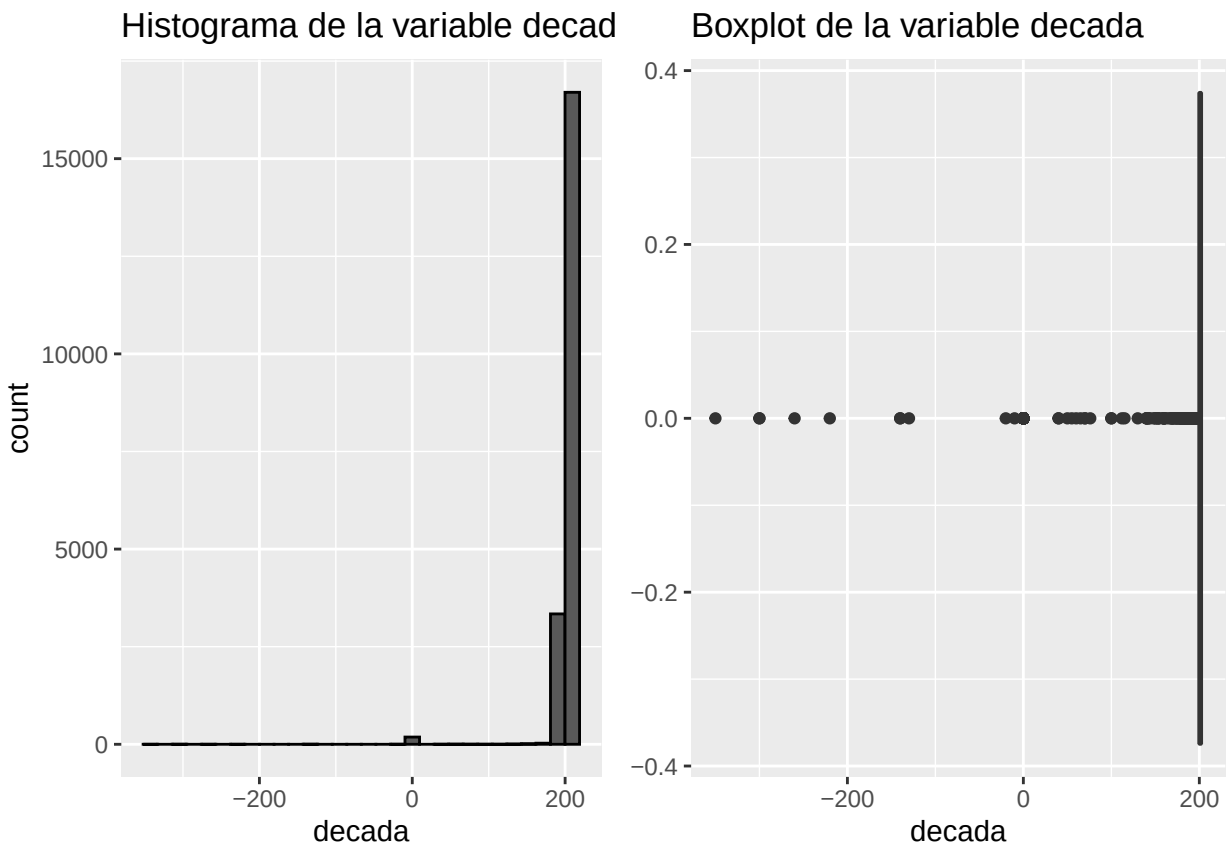
Se puede apreciar que los juegos anteriores a la década de los 70 son considerados todos como outliers. Igualmente se entiende que los valores a cero son por desconocerse la fecha de publicación del juego. Para visualizar mejor este caso se opta por hacer una combinación de dos gráficas (histograma y boxplot) que muestren la fecha de publicación de todos los juegos mediante la función grid.arrange de la librería **"gridEextra"**:

```
summary(datos$Year.Published)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   -3500    2001    2011    1984    2016    2022
```

```
library("ggplot2")
library("gridExtra")
g1 = ggplot(data = datos,aes(x = decada)) +
  geom_histogram(color="black") +
  labs(title="Histograma de la variable decada")

g2 = ggplot(data = datos,aes(x = decada)) +
  geom_boxplot()+
  labs(title="Boxplot de la variable decada")

grid.arrange(g1, g2, nrow=1)
```

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



Se confirma que la gran mayoría de juegos se han publicado a partir de finales del siglo XX.

Seguidamente mostramos los outliers de **Min.Players** y **Max.Players**, en el caso de los Max.Players se muestra igualmente una gráfica para confirmar que la mayoría de juegos tiene un Max.Players menor de 10. El caso de 0 jugadores en Max.Players se puede considerar como registro inválido por lo que procedemos a eliminarlos para que no afecten a los futuros modelos de correlación y regresiones.

```
unique(sort(boxplot.stats(datos$Min.Players)$out))
```

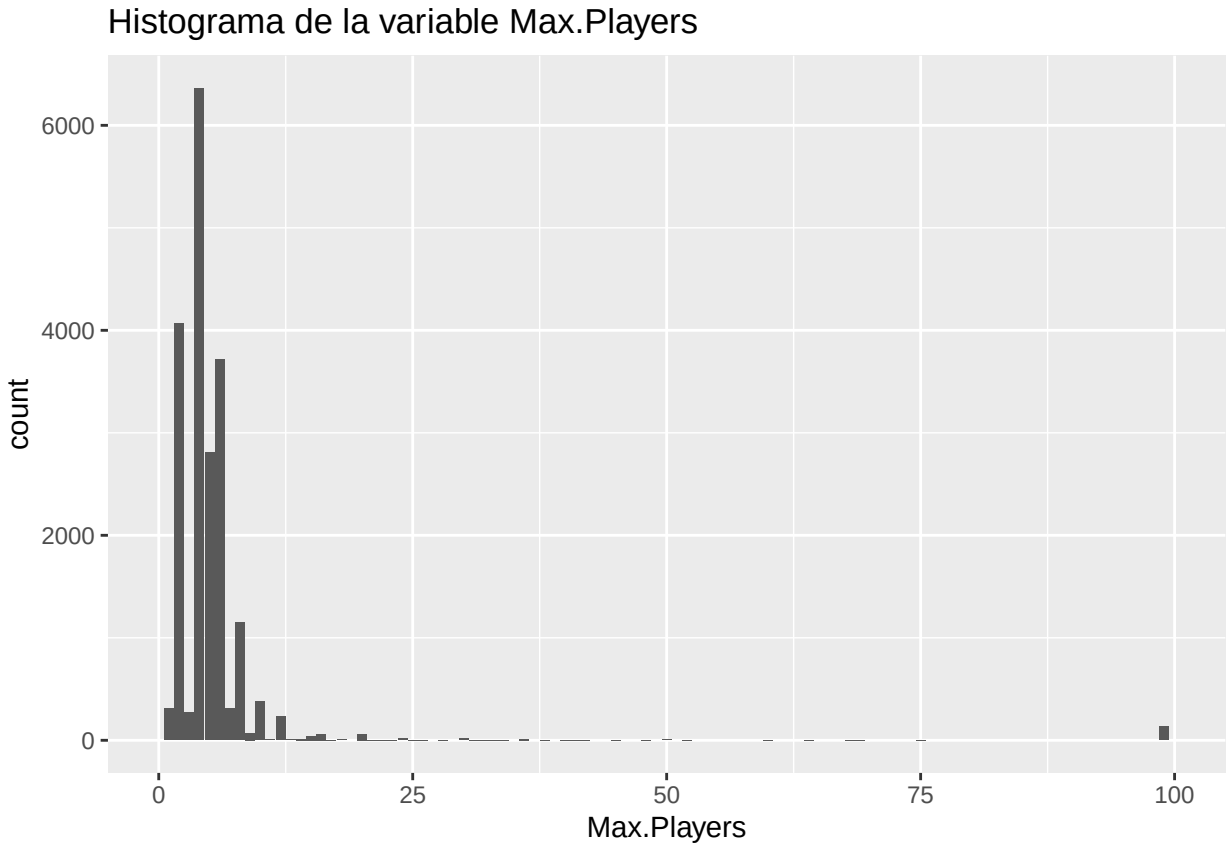
```
##  [1]  0  1  3  4  5  6  7  8  9 10
```

```
unique(sort(boxplot.stats(datos$Max.Players)$out))
```

```
##  [1]   0  10  11  12  13  14  15  16  17  18  20  21  22  23  24  25  26  28  30
## [20]  31  32  33  34  36  38  40  41  42  45  48  50  52  60  64  68  69  75  99
## [39] 100 120 127 163 200 362 999
```

```
# Eliminamos los registros con Max.Players = 0
datos <- datos[!(datos$Max.Players == 0),]
```

```
ggplot(datos, aes(Max.Players)) + geom_bar() + xlim(0,100)+
  ggtitle("Histograma de la variable Max.Players")
```



Mostramos los outliers de **Play.Time** y de **Min.Age**, en el caso de el tiempo de juego sorprende que haya juegos con un tiempo de cada partida de 1.000 horas (60.000 minutos).

```
unique(sort(boxplot.stats(datos$Play.Time)$out))
```

```
##  [1]   200   210   222   225   240   250   270   280   290   300   320   340
## [13]   360   400   420   450   480   500   540   600   660   700   720   750
## [25]   780   810   840   900   960   999  1000  1080  1200  1260  1440  1500
## [37]  1710  1740  1800  2100  2160  2400  2480  2880  3000  3600  3900  4000
## [49]  4200  4500  4560  5000  5400  6000  7920  8640 10000 12000 14400 17280
## [61] 22500 60000
```

```
unique(sort(boxplot.stats(datos$Min.Age)$out))
```

```
## [1]  0  1 21 25
```

En el caso de **User.Rated** hay casi 2.000 outliers, por lo que mostramos a modo de ejemplo simplemente el valor más bajo y alto.

```
min(sort(boxplot.stats(datos$Users.Rated)$out))
```

```
## [1] 889
```

```
max(sort(boxplot.stats(datos$Users.Rated)$out))
```

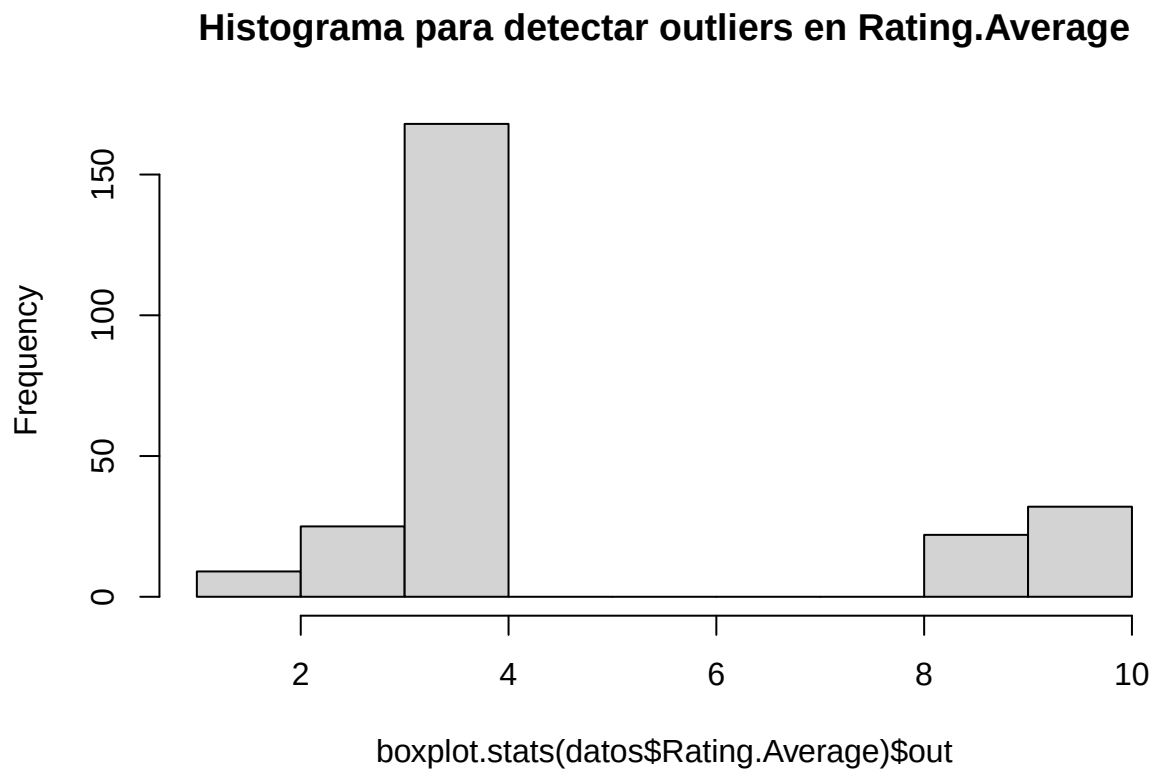
```
## [1] 102214
```

Con **Rating.Average** podemos observar como los valores inferiores a 4 o superiores a 8.85 son considerados como outliers, se muestra una gráfica de tipo histograma para facilitar su visualización:

```
unique(sort(boxplot.stats(datos$Rating.Average)$out))
```

```
##   [1] 1.05 1.10 1.32 1.43 1.50 1.54 1.55 1.78 1.90 2.06 2.14 2.23 2.24 2.28 2.34
##  [16] 2.43 2.50 2.63 2.64 2.68 2.71 2.74 2.75 2.80 2.83 2.85 2.86 2.87 2.93 3.00
##  [31] 3.02 3.03 3.04 3.09 3.10 3.11 3.17 3.18 3.19 3.21 3.22 3.23 3.24 3.27 3.29
##  [46] 3.31 3.32 3.34 3.35 3.36 3.38 3.39 3.41 3.42 3.45 3.47 3.48 3.50 3.51 3.52
##  [61] 3.53 3.54 3.55 3.56 3.57 3.59 3.61 3.62 3.64 3.65 3.66 3.67 3.68 3.69 3.70
##  [76] 3.71 3.72 3.73 3.74 3.76 3.77 3.78 3.79 3.80 3.81 3.82 3.83 3.84 3.85 3.86
##  [91] 3.87 3.88 3.89 3.90 3.91 3.93 3.94 3.95 3.96 3.97 3.98 3.99 4.00 8.85 8.87
## [106] 8.88 8.89 8.90 8.92 8.93 8.94 8.95 8.98 8.99 9.04 9.05 9.06 9.07 9.10 9.11
## [121] 9.12 9.13 9.14 9.16 9.18 9.19 9.21 9.22 9.23 9.24 9.25 9.31 9.34 9.43 9.46
## [136] 9.54 9.58
```

```
hist(boxplot.stats(datos$Rating.Average)$out,
     main="Histograma para detectar outliers en Rating.Average")
```



Para las variables **Complexity.Average** y **Owned.Users** se muestra solo el valor más bajo de sus respectivos outliers ya que todos los que están por encima también son considerados como valores extremos.

```
min(unique(sort(boxplot.stats(datos$Complexity.Average)$out)))
```

```
## [1] 4.38
```

```
min(unique(sort(boxplot.stats(datos$Owned.Users)$out)))
```

```
## [1] 1955
```

El caso del campo **Domains** es diferente, ya que es un campo alfanumérico que contiene varios valores para cada registro. Para hacernos una idea visualizamos los valores que contiene el campo sin previo tratamiento:

```
head(unique(datos$Domains),20)
```

```
##  [1] Strategy Games, Thematic Games     Strategy Games
##  [3] Thematic Games                     Strategy Games, Wargames
##  [5] Thematic Games, Wargames           Family Games, Strategy Games
##  [7] Customizable Games, Thematic Games Abstract Games, Family Games
##  [9] Customizable Games                 Family Games
## [11] Party Games                        Customizable Games, Wargames
## [13] Wargames                           Party Games, Thematic Games
## [15] Abstract Games                     Customizable Games, Strategy Games
## [17] Family Games, Thematic Games       Family Games, Party Games
## [19] Abstract Games, Strategy Games     Children's Games, Family Games
## 40 Levels: Abstract Games ... Wargames
```

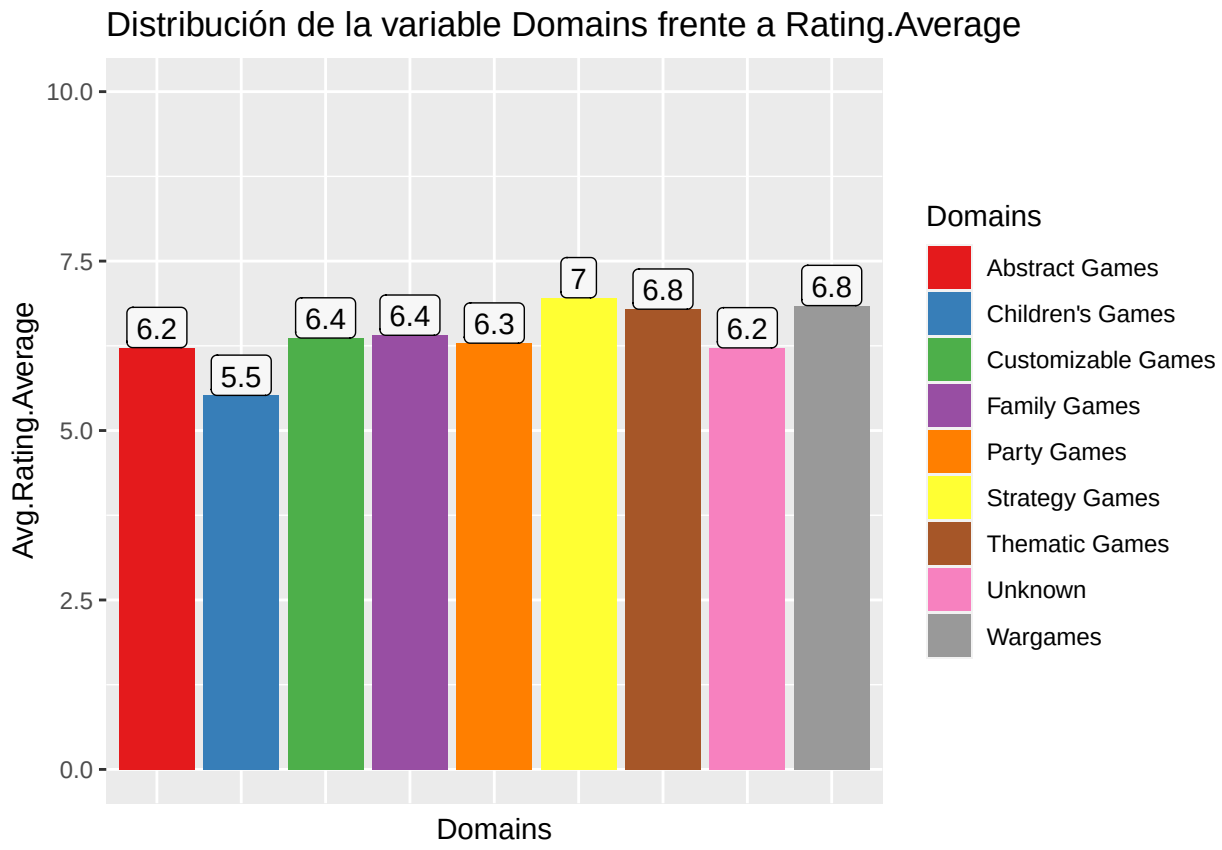
Para solucionarlo decidimos utilizar la librería **tidyr** para poder separar los valores y hacer un cálculo de la valoración media que dan los usuarios en función del género del juego (media de **Rating.Average** vs **Domains**).

```
# tidyr
library(tidyr)
Domains_separated = datos %>% separate_rows(Domains, sep = ", ")

# Agregamos datos
Domains_Rating.Average = aggregate(Domains_separated$Rating.Average,
                                   by=list(Domains=Domains_separated$Domains),
                                   FUN=mean
                                   )
# Renombramos columnas
colnames(Domains_Rating.Average) = c("Domains","Avg.Rating.Average")

# Mostramos de manera ordenada
Domains_Rating.Average[order(-Domains_Rating.Average$Avg.Rating.Average),]
```

```
##              Domains Avg.Rating.Average
## 6     Strategy Games           6.958352
## 9           Wargames           6.843323
## 7     Thematic Games           6.790000
## 4       Family Games           6.405332
## 3 Customizable Games           6.364392
```

```
## 5         Party Games           6.289003
## 1     Abstract Games           6.223663
## 8            Unknown           6.219168
## 2   Children's Games           5.518548
```

```
# Mostramos plot de la distribución
ggplot(Domains_Rating.Average, aes(x=Domains, y=Avg.Rating.Average, fill=Domains)) +
  geom_bar(stat = "identity") +
  ggtitle("Distribución de la variable Domains frente a Rating.Average") +
  scale_fill_brewer(palette = "Set1") +
  geom_label(aes(label=round(Avg.Rating.Average,1)), vjust=0, fill = "gray97") +
    theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) + scale_color_manual(values = col)+ ylim(0,10)
```



Puede apreciarse que los juegos mejor valorados son los de estrategia (7 sobre 10) y los peor valorados los infantiles (5.5 sobre 10)

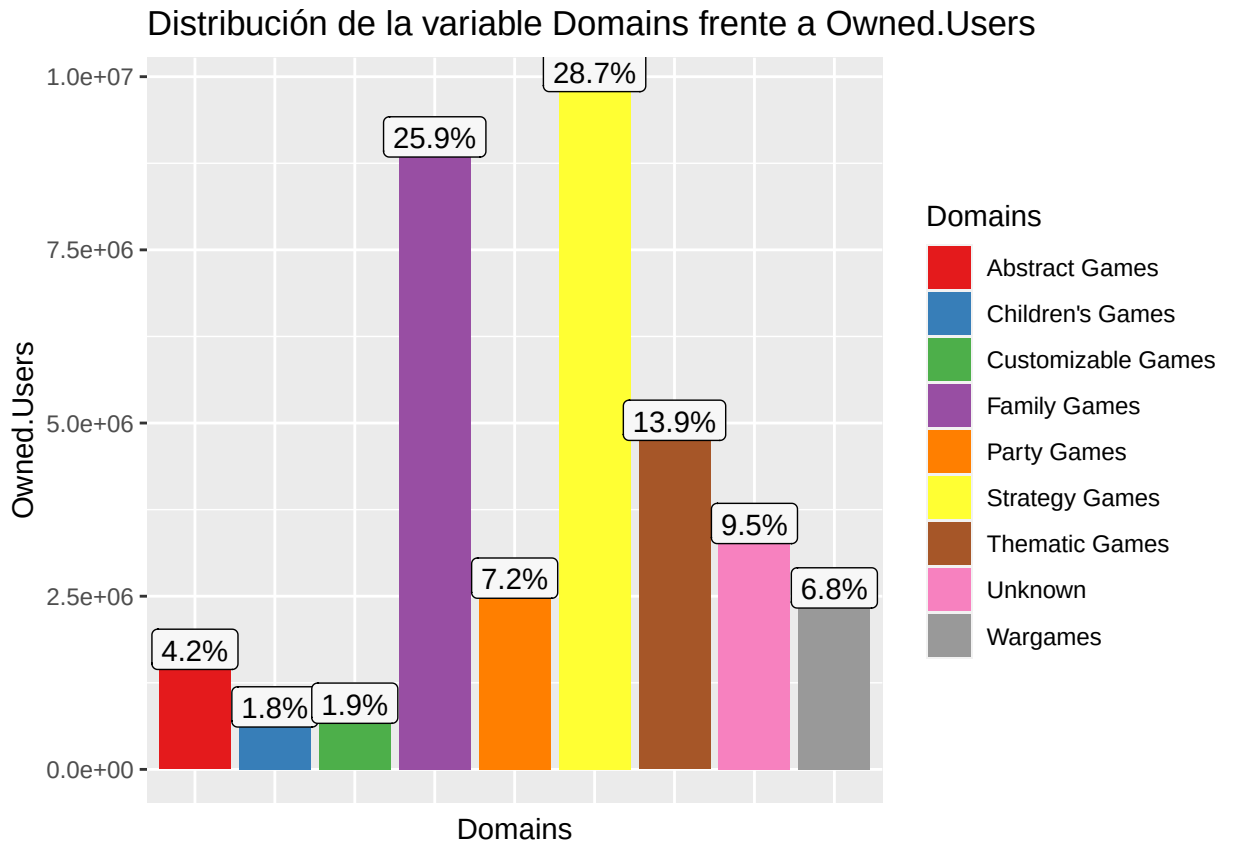
Realizamos un ejercicio similar para visualizar la distribución de la variable **Domains** frente al sumatorio de **Owned.Users** con el objetivo de identificar que tipo de juegos son los más comprados:

```
# tidyr
library(tidyr)
Domains_separated = datos %>% separate_rows(Domains, sep = ", ")

# Agregamos datos
Domains_Owned.Users = aggregate(Domains_separated$Owned.Users,
                                by=list(Domains=Domains_separated$Domains),
```

```
                                FUN=sum
                                )
# Renombramos columnas
colnames(Domains_Owned.Users) = c("Domains","Total.Owned.Users")

# Muestra ordenada
Domains_Owned.Users[order(-Domains_Owned.Users$Total.Owned.Users),]
```

```
##              Domains Total.Owned.Users
## 6     Strategy Games           9784440
## 4       Family Games           8840076
## 7     Thematic Games           4748879
## 8            Unknown           3259253
## 5        Party Games           2470187
## 9           Wargames           2329079
## 1     Abstract Games           1442373
## 3 Customizable Games            664672
## 2   Children's Games            611350
```

```
datos_agregados = aggregate(Domains_separated$Owned.Users,
          by=list(Domains_separated$Domains),
          FUN=sum)

# Renombramos columnas
colnames(datos_agregados) = c("Domains","Owned.Users")

# Creamos campo nuevo con los porcentajes respecto al total para facilitar el plot
datos_agregados$pct = datos_agregados$Owned.Users / sum(datos_agregados$Owned.Users)

# library scales para visualizar mejor los porcentajes
library('scales')

# Mostramos plot de la distribución de Domains vs Owned.Users
ggplot(datos_agregados, aes(x=Domains, y=Owned.Users, fill=Domains)) +
  geom_bar(stat = "identity") +
  ggtitle("Distribución de la variable Domains frente a Owned.Users") +
  scale_fill_brewer(palette = "Set1") +
  geom_label(aes(label=percent(datos_agregados$pct, accuracy=0.1)),
             vjust=0, fill = "gray97") +
  theme(axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) + scale_color_manual(values = col)
```



Finalmente realizamos un estudio del campo **Mechanics**, que al igual que **Domains** es multivalor, aunque en este caso con un volumen muy superior de diferentes valores. Con el objeto de visualizar las principales mecánicas de una forma alternativa se opta por mostrarlo en una nube de palabras

```
# tidyr
library(tidyr)

Mechanics_separated = datos %>% separate_rows(Mechanics, sep = ", ")

# Visualizamos las 20 principales mecánicas
head(sort(table(Mechanics_separated$Mechanics), decreasing = T),20)
```

```
##
##                  Dice Rolling              Hand Management
##                          5621                         4140
##               Set Collection       Variable Player Powers
##                          2717                         2515
##                  Hexagon Grid                   Simulation
##                          2233                         1928
##                 Card Drafting               Tile Placement
##                          1725                         1684
##                Modular Board                      Unknown
##                          1619                         1540
##     Area Majority / Influence                Grid Movement
##                          1500                         1473
##             Cooperative Game          Roll / Spin and Move
##                          1397                         1292
```

```
##                 Area Movement Simultaneous Action Selection
##                          1116                          1097
##                 Action Points               Auction/Bidding
##                          1074                          1051
##                     Take That               Team-Based Game
##                           983                           932
```

```
Mechanics_separated = data.frame(head(sort(table(Mechanics_separated$Mechanics),
                                           decreasing = T),40))

# Nube de palabras con las principales mecánicas
library(wordcloud2)
wordcloud2(data=Mechanics_separated, size=0.4, color='random-dark', shape = "circle")
```

Análisis de los datos

Selección de los grupos de datos que se quieren analizar/comparar (planificación de los análisis a aplicar)

Para llevar a cabo el análisis de los datos y con motivo de responder las preguntas que nos realizábamos en el primer apartado, se plantea la elección de las variables con las que vamos a trabajar.

Para ver cómo se distribuye la muestra, vamos a comprobar la normalidad de todas las variables. Posteriormente, realizaremos los agrupamientos que creamos necesarios.

Comprobación de la normalidad y homogeneidad de la varianza.

Las variables cuantitativas de nuestro estudio no siguen una distribución normal multivariante. Para verlo, haremos uso de la prueba de normalidad de Anderson-Darling:

```
# Comprobamos la normalidad
library(nortest)
alpha = 0.05
col.names = colnames(datos)
for (i in 1:ncol(datos)) {
  if (i == 1) cat("Variables que no siguen una distribución normal:\n")
    if (is.integer(datos[,i]) | is.numeric(datos[,i])) {
    p_val = ad.test(datos[,i])$p.value
    if (p_val < alpha) {
    cat(col.names[i])
    # Format output
    if (i < ncol(datos) - 1) cat(", ")
    if (i %% 1 == 0) cat("\n")
    }
  }
}
```

```
## Variables que no siguen una distribución normal:
## ID,
## Year.Published,
## Min.Players,
## Max.Players,
## Play.Time,
## Min.Age,
## Users.Rated,
## Rating.Average,
## Complexity.Average,
## Owned.Users,
## decada
```

Una vez hemos comprobado la no normalidad de la población, pasamos a estudiar mediante el test no paramétrico de Fligner-Killeen la homogeneidad de la varianza. En este caso, plantearemos tantas hipótesis nulas como variables vayamos a estudiar:

```
# Realizamos Test de Flinger para analizar la homogeneidad de la varianza
#  de los grupos de variables pre-seleccionados.
fligner.test(Year.Published ~ Play.Time, data = datos) # ANALIZAR
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Year.Published by Play.Time
## Fligner-Killeen:med chi-squared = 1380.5, df = 115, p-value < 2.2e-16
```

```
fligner.test(Year.Published ~ Rating.Average, data = datos) # ANALIZAR
```

```
##
```

```
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Year.Published by Rating.Average
## Fligner-Killeen:med chi-squared = 2048.9, df = 619, p-value < 2.2e-16
```

```
fligner.test(Year.Published ~ Owned.Users, data = datos) # ANALIZAR
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Year.Published by Owned.Users
## Fligner-Killeen:med chi-squared = 5509.6, df = 3995, p-value < 2.2e-16
```

```
fligner.test(Max.Players ~ Owned.Users, data = datos) # ANALIZAR
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Max.Players by Owned.Users
## Fligner-Killeen:med chi-squared = 4282.8, df = 3995, p-value =
## 0.0008078
```

```
fligner.test(Rating.Average ~ Owned.Users, data = datos)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Rating.Average by Owned.Users
## Fligner-Killeen:med chi-squared = 5464.2, df = 3995, p-value < 2.2e-16
```

```
# Todos los p-value son inferiores a 0.05 por lo que la hipótesis de que
#  las varianzas de ambas muestras es que no son homogéneas.
```

Obsérvese que en los cuatro casos el p-valor es inferior a 0.05, por lo que rechazamos la hipótesis nula de homocedasticidad.

Aplicación de pruebas estadísticas para comparar los grupos de datos. En función de los datos y el objetivo del estudio, aplicar pruebas de contraste de hipótesis, correlaciones, regresiones, etc. Aplicar al menos tres métodos de análisis diferentes

Método 1: Correlación

En primer lugar, vamos a realizar un análisis de correlación entre las distintas variables cuantitativas de nuestro estudio. En particular, esto nos arrojará información acerca de la influencia de las variables independientes sobre la variable *Rating.Average*.

```r
# creamos copia del dataset para las pruebas
datos_pruebas <- datos[,c("Min.Players", "Max.Players", "Play.Time", "Min.Age",
                          "Users.Rated", "Year.Published","Complexity.Average",
                          "Owned.Users", "Rating.Average")]

corr_matrix <- matrix(nc = 2, nr = 0)
colnames(corr_matrix) <- c("estimate", "p-value")
# Calcular el coeficiente de correlación para cada variable cuantitativa
# con respecto al campo "precio"
for (i in 1:(ncol(datos_pruebas) - 1)) {
  if (is.integer(datos_pruebas[,i]) | is.numeric(datos_pruebas[,i])) {
    spearman_test = cor.test(datos_pruebas[,i],
    datos_pruebas[,length(datos_pruebas)],
    method = "spearman",
    #Hemos incluido exact=FALSE
    exact = FALSE)
    corr_coef = spearman_test$estimate
    p_val = spearman_test$p.value
    # Add row to matrix
    pair = matrix(ncol = 2, nrow = 1)
    pair[1][1] = corr_coef
    pair[2][1] = p_val
    corr_matrix <- rbind(corr_matrix, pair)
    rownames(corr_matrix)[nrow(corr_matrix)] <- colnames(datos_pruebas)[i]
    }
}
print(corr_matrix)
```

```
##                      estimate       p-value
## Min.Players        -0.2080513 9.710286e-196
## Max.Players        -0.1932434 1.225015e-168
## Play.Time           0.3630873  0.000000e+00
## Min.Age             0.2867929  0.000000e+00
## Users.Rated         0.2536687 3.830793e-293
## Year.Published      0.4248753  0.000000e+00
## Complexity.Average  0.5061303  0.000000e+00
## Owned.Users         0.2758823  0.000000e+00
```

Nótese que todas las variables presentan en mayor o menor medida una clara influencia sobre la variable *Rating.Average*. De aquí podemos obtener dos claras conclusiones:

- La complejidad del juego es, a priori, lo que más influye a la hora de realizar una valoración personal.
- El año en que fue distribuido presenta una gran importancia a la hora de realizar la evaluación.

Ahora bien, si queremos ver qué influencia tienen unas variables cuantitativas sobre las otras, podemos verlo mediante la siguiente representación:

```r
library(corrplot)
```

```
## corrplot 0.92 loaded
```

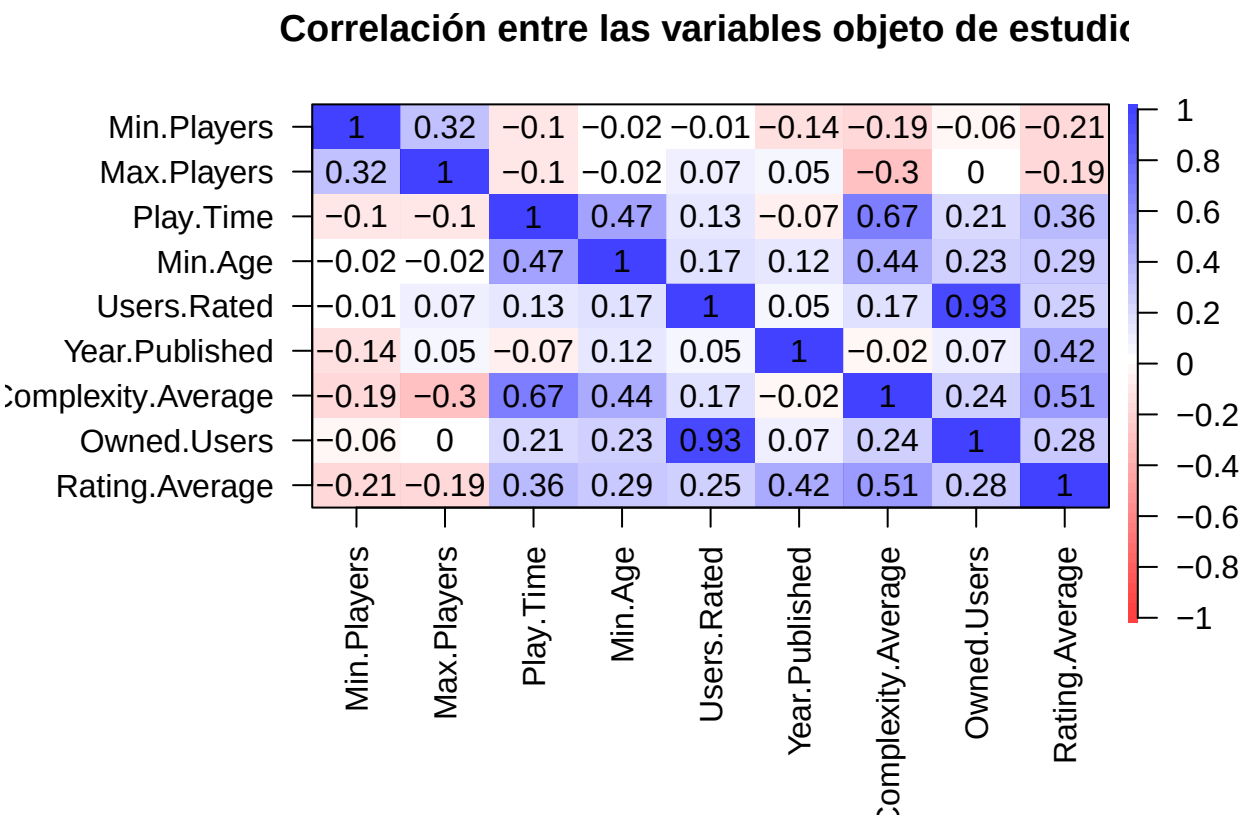
```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following objects are masked from 'package:scales':
##
##     alpha, rescale
```

```
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
cor <- cor(datos_pruebas,method = "spearman")

# Mostrámos matriz de correlación entre variables
corPlot(cor, xlas=2, main = "Correlación entre las variables objeto de estudio")
```



Puede apreciarse una fuerte correlación entre las variables *Users.Rated* y *Owned.Users*. Lo cual parece evidente, puesto que cuantas más personas tengan el juego, más podrán evaluarlo. Otras relaciones que arrojan bastante información son:

- Cuanto más complicado es el juego, más tiempo requiere.
- La complejidad del juego o el tiempo requerido para el mismo, influyen en la edad mínima recomendada.

Método 2: Contrastes de hipótesis

En segundo lugar, vamos a realizar un contraste de hipótesis para evaluar la influencia del tipo de juego seleccionado (infantiles o familiares) en la cantidad de usuarios que lo poseen.

La hipótesis nula consistirá en que estos dos tipos de juegos se venden en la misma medida. Esto ya ha sido probado mediante el análisis exploratorio, aún así lo veremos estadísticamente.

Debemos tener en cuenta que para poder utilizar este test los datos deberían estar distribuidos normalmente. Sin embargo, al ser n $>$ 30 los resultados arrojados por este test son igualmente válidos.

```
domain_children <-
datos[datos$Domains == "Children's Games",]$Owned.Users
domain_family <-
datos[datos$Domains == "Family Games",]$Owned.Users

t.test(domain_children, domain_family,
alternative = "less")
```

```
##
##  Welch Two Sample t-test
##
## data:  domain_children and domain_family
## t = -12.945, df = 1405.6, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##       -Inf -2953.307
## sample estimates:
## mean of x mean of y
##  540.4057 3923.9350
```

Como el p-valor es menor que el valor de significación fijado, rechazamos la hipótesis nula. Por lo tanto, se induce que los juegos de familiares se venden con mayor facilidad que los juegos infantiles.

Método 3: Regresión

Por último, se plantea un modelo de regresión lineal en el que usaremos únicamente las variables cuantitativas.

```
datos_regresion <- datos[,c("Year.Published","Min.Players", "Max.Players",
                            "Play.Time", "Min.Age", "Users.Rated", "Rating.Average",
                            "Complexity.Average", "Owned.Users")]
```

Se realizan 3 modelos diferentes con el objetivo de realizar una estimación de la variable Owned.Users, dando como resultado que los modelos 2 y 3 son los que mas se ajustan con unas puntuaciones de 0.9722 y 0.9725 respectivamente. Esto se debe a la fuerte correlación entre Owned.Users y Users.Rated.

```
V.Year.Published <- datos_regresion$Year.Published
V.Min.Players <- datos_regresion$Min.Players
V.Max.Players <- datos_regresion$Max.Players
V.Play.Time <- datos_regresion$Play.Time
V.Min.Age <- datos_regresion$Min.Age
V.Users.Rated <- datos_regresion$Users.Rated
V.Complexity.Average <- datos_regresion$Complexity.Average
```

```
V.Owned.Users <- datos_regresion$Owned.Users
V.Rating.Average <- datos_regresion$Rating.Average

modelo1 <- lm(V.Owned.Users ~ V.Year.Published + V.Min.Players + V.Max.Players +
                V.Play.Time + V.Min.Age + V.Rating.Average, data = datos_regresion)

modelo2 <- lm(V.Owned.Users ~ V.Users.Rated, data = datos_regresion)

modelo3 <- lm(V.Owned.Users ~ V.Year.Published + V.Min.Players + V.Max.Players +
                V.Play.Time + V.Min.Age + V.Rating.Average + V.Complexity.Average +
                V.Users.Rated, data = datos_regresion)
```

```
summary(modelo1)$r.squared
```

```
## [1] 0.03447946
```

```
summary(modelo2)$r.squared
```

```
## [1] 0.9722581
```

```
summary(modelo3)$r.squared
```

```
## [1] 0.9725607
```

Resolución del problema. A partir de los resultados obtenidos, ¿cuáles son las conclusiones? ¿Los resultados permiten responder al problema?

Se procede a responder a las preguntas inicialmente propuestas:

-¿Cuales son los géneros (Domains) más vendidos (Owned.Users)?

Como pudimos comprobar anteriormente mediante análisis exploratorio, los géneros más vendidos son en primer lugar los de estrategia con cerca de 10^7 Owned.Users, seguido de los juegos familiares con cerca de 9*10^6 Owned.Users.

Igualmente se realizó un contraste de hipótesis entre dos de los principales géneros y se confirmó estadísticamente que las clases de Domains influyen en Owned.Users.

-¿Cuales son las mecánicas (Mechanics) mas comunes?

Las mecánicas más populares son Dice Rolling, Hand Management y Set Collection, del total de las aproximadamente 200 mecánicas existentes.

-¿Influye el género (Domains) en la valoración media del juego (Rating.Average)?

La segunda gráfica de barras de distribución de la variable domains frente a Rating Average nos indica que los juegos de estrategia (7 sobre 10), de guerra y temáticos (6,8 sobre 10 ambos) son los mejor valorados. Llama la atención como los juegos familiares, a pesar de ser el segundo género más vendido, tiene una puntuación normal.

-¿Qué factores influyen más en la duración de las partidas (Play.Time)?

Por medio del análisis de correlación se ha podido comprobar que las variables que mas afectan a la duración de las partidas son la complejidad del juego *(complexity.Average)* y la edad mínima *(Min.Age)*

-¿El año de publicación influye en alguna otra variable?

El análisis de correlación indica que solo influye en una variable, y es la de la valoración media *(Rating.Average)*.

-¿Qué sería necesario para hacer una estimación de las ventas?

El análisis de regresión nos demuestra que para realizar la mejor estimación posible de ventas de un juego vamos a necesitar disponer de la cantidad de usuarios que han valorado el juego (V.Users.Rated). Sin ese dato la estimación tendría un margen de error muy elevado.

Repositorio Git y Vídeo

El enlace con los documentos, código y dataset se encuentran en la siguiente dirección:

https://github.com/eliaspaez/MDS-TYCDD-PRA2

El enlace al vídeo se encuentra en:

https://drive.google.com/file/d/1NZCXmNyOCfoGUEEE8tkxejZqfSOCKoEW/view?usp=sharing

El enunciado de la práctica pide tanto el csv con los datos originales como un csv con los datos finales analizados, por lo que se procede a volcar el dataset a un nuevo csv, que será subido al repositorio Git igualmente:

```
write.table (datos, file ="BGG_Data_Set_final.csv", append=F, quote=F, sep='\t',
             row.names=F, col.names=T, fileEncoding="UTF-8")
```

Contribuciones y Participantes

Contribuciones	Firma
Investigación previa	Juan Miguel Iglesias Labraca y Elías Páez de la Rosa
Redacción de las respuestas	Juan Miguel Iglesias Labraca y Elías Páez de la Rosa
Desarrollo código	Juan Miguel Iglesias Labraca y Elías Páez de la Rosa

BIBLIOGRAFÍA

- Subirats, L., Pérez, D., Calvo, M. (2019). Introducción al ciclo de vida de los datos. Editorial UOC.
- Subirats, L., Pérez, D., Calvo, M. (2019). Introducción a la limpieza y análisis de los datos. Editorial UOC. Megan Squire (2015). Clean Data. Packt Publishing Ltd.
- Jiawei Han, Micheine Kamber, Jian Pei (2012). Data mining: concepts and techniques. Morgan Kaufmann.
- Jason W. Osborne (2010). Data Cleaning Basics: Best Practices in Dealing with Extreme Scores. Newborn and Infant Nursing Reviews; 10 (1): pp. 1527-3369.
- Peter Dalgaard (2008). Introductory statistics with R. Springer Science & Business Media.
- Wes McKinney (2012). Python for Data Analysis. O'Reilley Media, Inc.
- Tutorial de Github (GitHub Guides website). https://guides.github.com/activities/hello-world