PRA 2 - Tipología y ciclo de vida de los datos

2023-01-08

Introducción

Debido a que los datos que extrjimos meidante *Web Scrapping* no son los más adecuados para realizar una regresión, es por esllo que hemos decidido utilizar una muestra de *Kaggle*, más concretamente, Board Games, este conjunto de datos contiene la información de los juegos de mesa de la web BoardGameGeek a fecha de enero de 2021.

Presentamos los primeros cinco registros de nuestra muestra.

```
dt <- read.csv2('bgg_dataset.csv', sep = ';') %>% data.table()
head(dt, 5)
```

```
##        ID                              Name Year.Published Min.Players
## 1: 174430                        Gloomhaven           2017           1
## 2: 161936         Pandemic Legacy: Season 1           2015           2
## 3: 224517                 Brass: Birmingham           2018           2
## 4: 167791                 Terraforming Mars           2016           1
## 5: 233078 Twilight Imperium: Fourth Edition           2017           3
##    Max.Players Play.Time Min.Age Users.Rated Rating.Average BGG.Rank
## 1:           4       120      14       42055           8.79        1
## 2:           4        60      13       41643           8.61        2
## 3:           4       120      14       19217           8.66        3
## 4:           5       120      12       64864           8.43        4
## 5:           6       480      14       13468           8.70        5
##    Complexity.Average Owned.Users
## 1:               3.86       68323
## 2:               2.84       65294
## 3:               3.91       28785
## 4:               3.24       87099
## 5:               4.22       16831
##
## 1: Action Queue, Action Retrieval, Campaign / Battle Card Driven, Card Play Conflict Resolution, Comm
## 2:
## 3:
## 4:
## 5:
##                          Domains
## 1: Strategy Games, Thematic Games
## 2: Strategy Games, Thematic Games
## 3:                 Strategy Games
## 4:                 Strategy Games
## 5: Strategy Games, Thematic Games
```

Descripción del dataset

Las variables que tenemos en nuestro conjunto de datos son las siguientes

- **ID**: Identificación del juego.
- **Name**: Nombre del juego.
- **Year.Published**: Año de publicación.
- **Min.Players**: Número mínimo de jugadores.
- **Max.Players**: Número máximo de jugadores.
- **Play.Time**: Duración estimada de cada partida.
- **Min.Age**: Edad mínima recomendada para el juego.
- **Users.Rated**: Número 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.).

Explotar estos datos puede tener como objetivo aumentar los beneficios de una tienda de juegos de mesa, por lo que algunas de las principales preguntas a las que se buscará respuesta son

- *¿Qué tipo de juegos son los que más se venden?*
- *¿Las valoraciones positivas influyen en el número de ventas?*
- *¿La duración de las partidas afecta a las ventas?*
- *¿Es posible estimar las ventas que tendrá un juego de mesa a partir de los datos que disponemos?*

Estas son algunas de las preguntas que nos han surgido, a lo largo de esta práctica intentaremos dar una respuesta a estas preguntas y a otras que puedan surgir.

Integración y selección de los datos de interés a analizar

A continuación mostramos como se distribuyen los datos que tenemos

```
summary(dt)
```

```
##        ID              Name           Year.Published   Min.Players
##  Min.   :     1   Length:20343       Min.   :-3500   Min.   : 0.00
##  1st Qu.: 11029   Class :character   1st Qu.: 2001   1st Qu.: 2.00
##  Median : 88931   Mode  :character   Median : 2011   Median : 2.00
##  Mean   :108216                      Mean   : 1984   Mean   : 2.02
##  3rd Qu.:192940                      3rd Qu.: 2016   3rd Qu.: 2.00
##  Max.   :331787                      Max.   : 2022   Max.   :10.00
##  NA's   :16                          NA's   :1
##   Max.Players        Play.Time          Min.Age        Users.Rated
##  Min.   :  0.000   Min.   :    0.00   Min.   : 0.000   Min.   :    30
##  1st Qu.:  4.000   1st Qu.:   30.00   1st Qu.: 8.000   1st Qu.:    55
##  Median :  4.000   Median :   45.00   Median :10.000   Median :   120
##  Mean   :  5.672   Mean   :   91.29   Mean   : 9.601   Mean   :   841
##  3rd Qu.:  6.000   3rd Qu.:   90.00   3rd Qu.:12.000   3rd Qu.:   385
##  Max.   :999.000   Max.   :60000.00   Max.   :25.000   Max.   :102214
##
##  Rating.Average     BGG.Rank     Complexity.Average  Owned.Users
##  Min.   :1.050   Min.   :    1   Min.   :0.000      Min.   :     0
```

```
##  1st Qu.:5.820   1st Qu.: 5088   1st Qu.:1.330      1st Qu.:   146
##  Median :6.430   Median :10173   Median :1.970      Median :   309
##  Mean   :6.403   Mean   :10173   Mean   :1.991      Mean   :  1408
##  3rd Qu.:7.030   3rd Qu.:15258   3rd Qu.:2.540      3rd Qu.:   864
##  Max.   :9.580   Max.   :20344   Max.   :5.000      Max.   :155312
##                                                     NA's   :23
##   Mechanics           Domains
##  Length:20343       Length:20343
##  Class :character   Class :character
##  Mode  :character   Mode  :character
##
##
##
##
```

De esta manera tenemos una breve idea de como son los datos que disponemos. También nos sirve para decidir si descartmos previamente alguna variable que no vaya a aportarnos ninguna información, como es el caso de las variables *ID*, *Name* y *BGG.Rank* las cuales no aportan información relevante a nuestros. También descartamos la variable *Mechanics* ya que requiere de un tratamiento especial si queremos aprovecharla.

```
dt <- dt %>% dplyr::select(-c(ID, Name, BGG.Rank, Mechanics))
```

Limpieza de los datos

Ahora que ya disponemos de un conjunto de datos inicial, el cual tendremos que explotar, necesitaremos limpiarlo ya que esto nos permitirá obtener mejores resultados posteriormente.

Tratamiento de los vacíos

Comenzaremos identificando los registros que contienen vacíos

```
apply(dt, 2, function(x){round((sum(is.na(x))/length(x))*100, 2)})
```

```
##     Year.Published        Min.Players        Max.Players          Play.Time
##               0.00               0.00               0.00               0.00
##            Min.Age       Users.Rated     Rating.Average Complexity.Average
##               0.00               0.00               0.00               0.00
##        Owned.Users            Domains
##               0.11               0.00
```

Como se puede apreciar la variable *Owned.Users* contiene un 0.11% de registros vacíos, debido a que esta cantidad es insignificate optaremos por prescindir de estos registros y por tanto los eliminaremos

```
dt <- dt %>% filter(!is.na(Owned.Users))
```

Buscaremos ahora vacíos en los datos que no se expresen como *NA*.

```
apply(dt, 2, function(x){round((sum(x=='')/length(x))*100, 2)})
```

```
##     Year.Published        Min.Players        Max.Players          Play.Time
##               0.00               0.00               0.00               0.00
##            Min.Age       Users.Rated     Rating.Average Complexity.Average
##               0.00               0.00               0.00               0.00
##        Owned.Users            Domains
##               0.00              49.88
```

Debido a que en esta ocasión encontramos una cantidad de registros vacíos en la varible *Domains* optamos en esta ocasión por asignar un valor a dichos valores vacíos.

```
dt[Domains == '', Domains := 'Other']
```

Al asignar el valor *Other* a nuestras variables, tenemos que dichas variables ahora cuentan con una categoría nueva, la cual nos permite identificar los registros de los cuales no teniamos información en un principio.

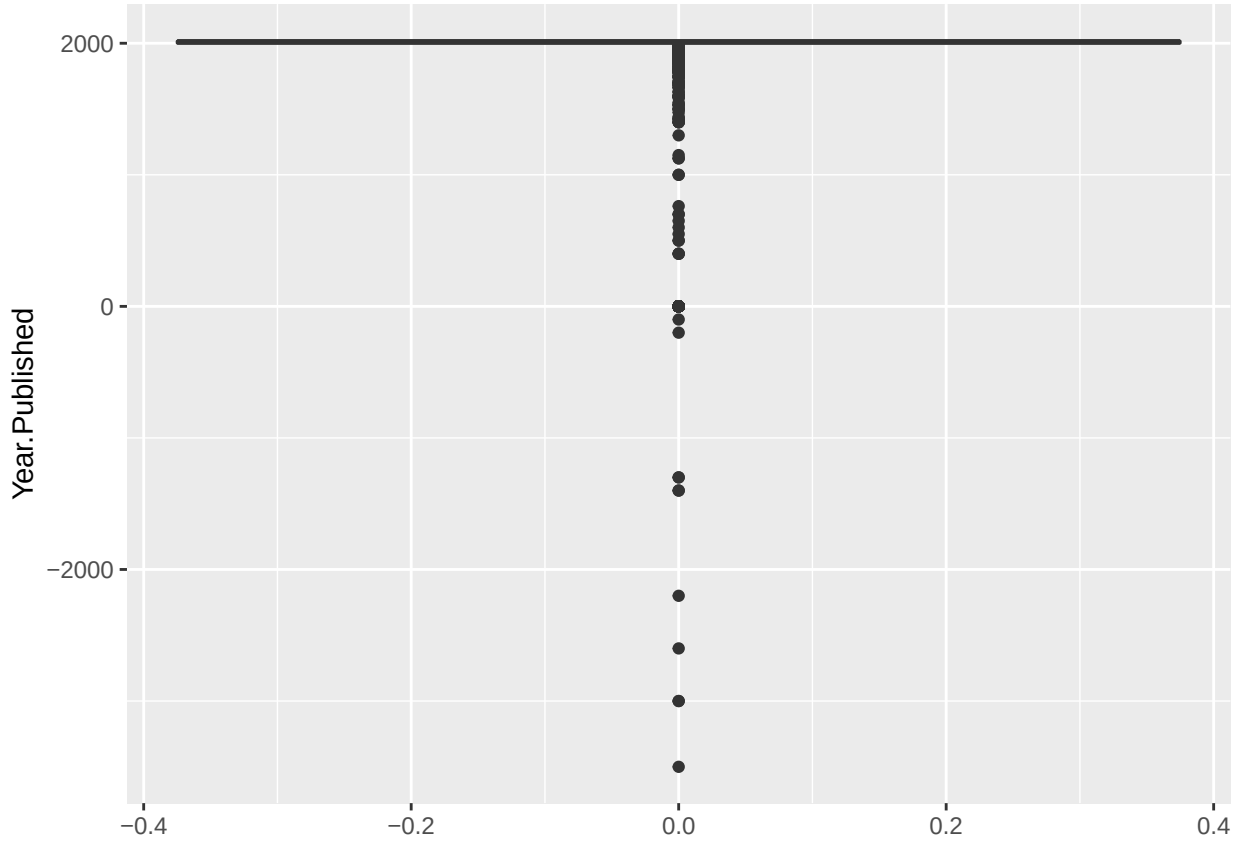
Después de esta transformación es muy poco probable que las variables de tipo categórico sufran más modificaciones, por lo que sería de gran utilidad transformarlas a tipo factor, ya que será algo que necesitemos más adelante

```
dt <- dt %>% separate_rows(Domains, sep = ', ') %>% data.table()
dt <- dt %>%
  mutate(Domains = as.factor(Domains))
```

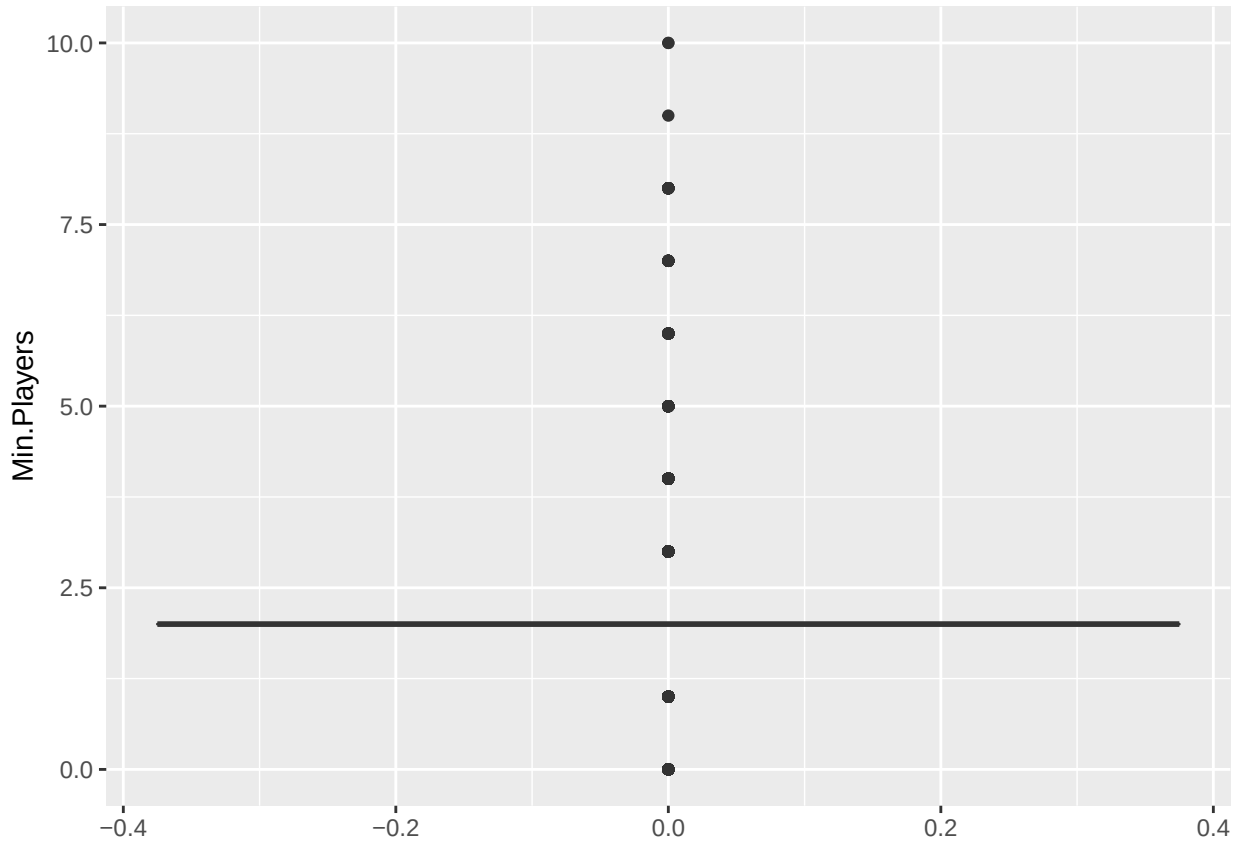
Tratamiento de outlayers

A continuación realizaremos un tratamiento para los distintos valores extremos que encontremos en nuestras varibles, es por ello que tenemos que identificar en nuestras variables númericas este tipo de valores.

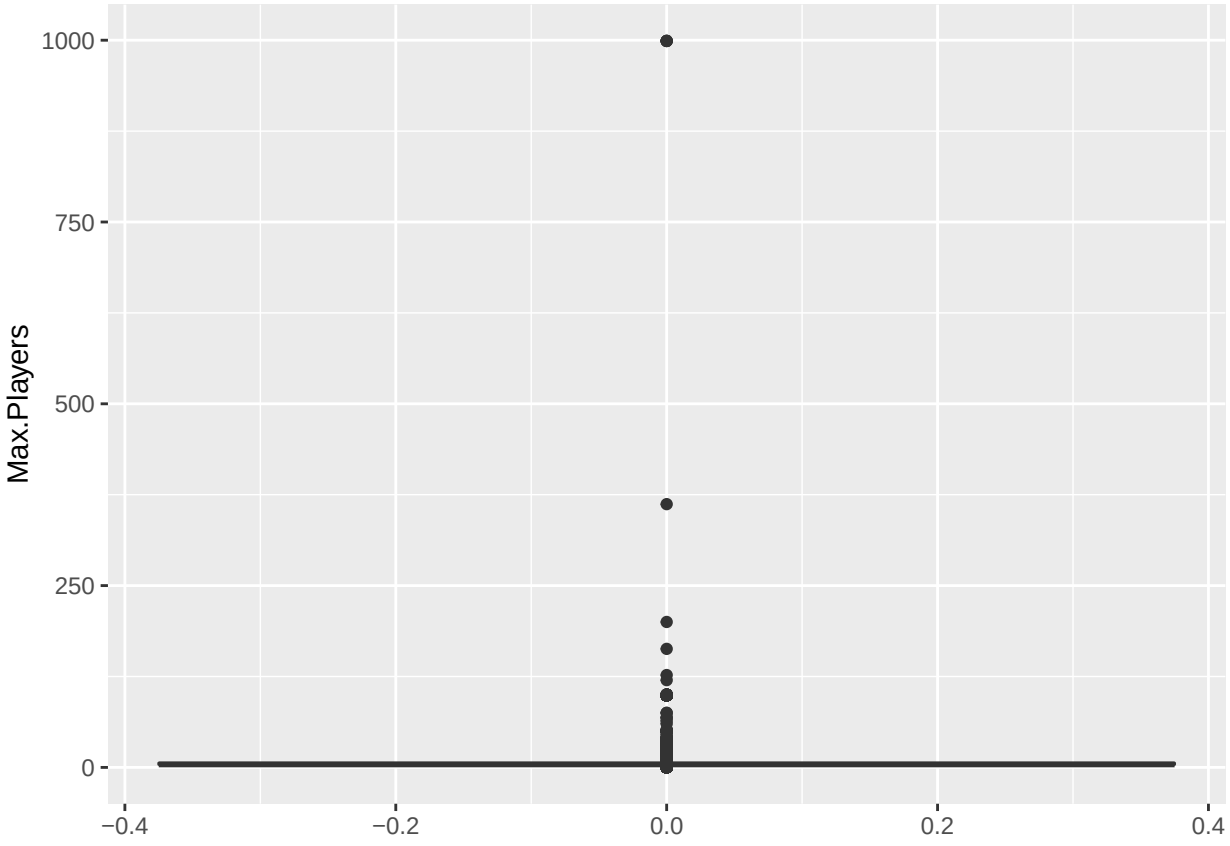
```
ggplot(dt, aes(y = Year.Published)) + geom_boxplot()
```



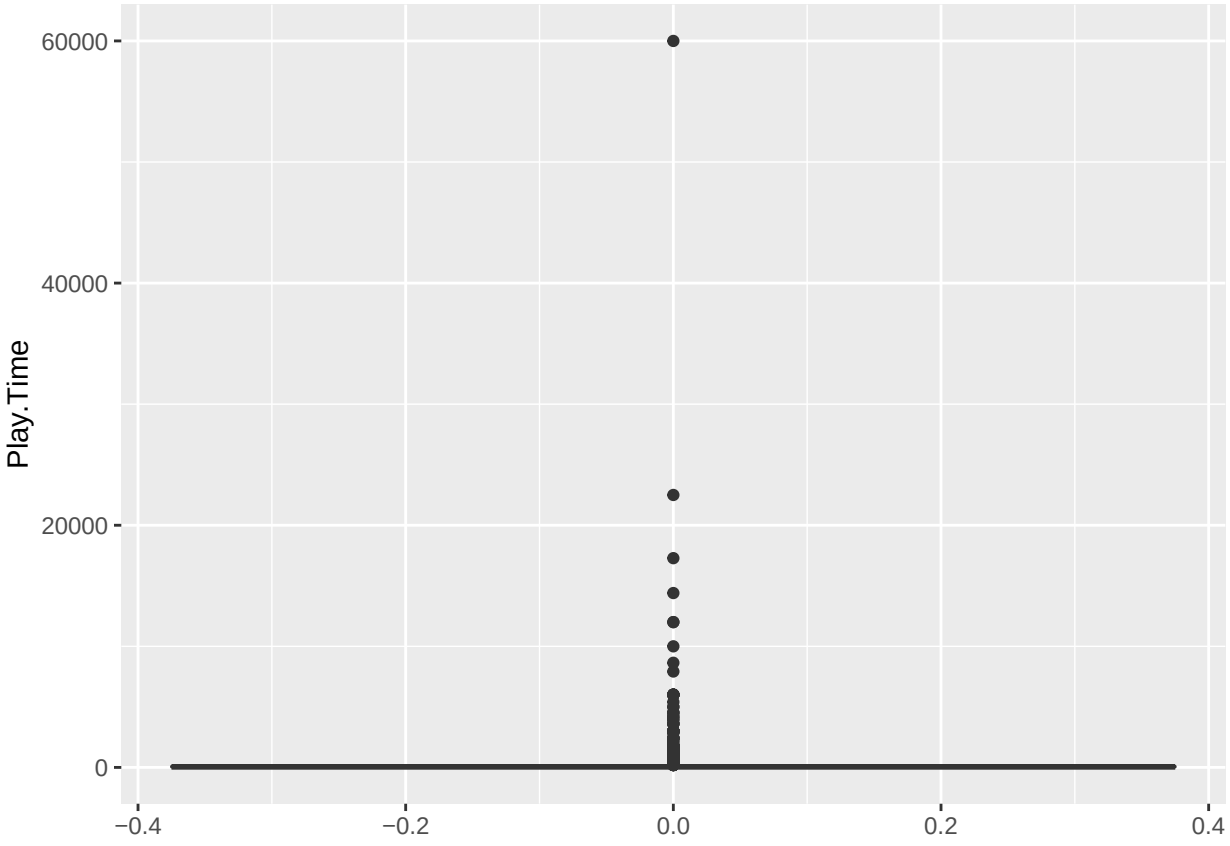

```
ggplot(dt, aes(y = Min.Players)) + geom_boxplot()
```



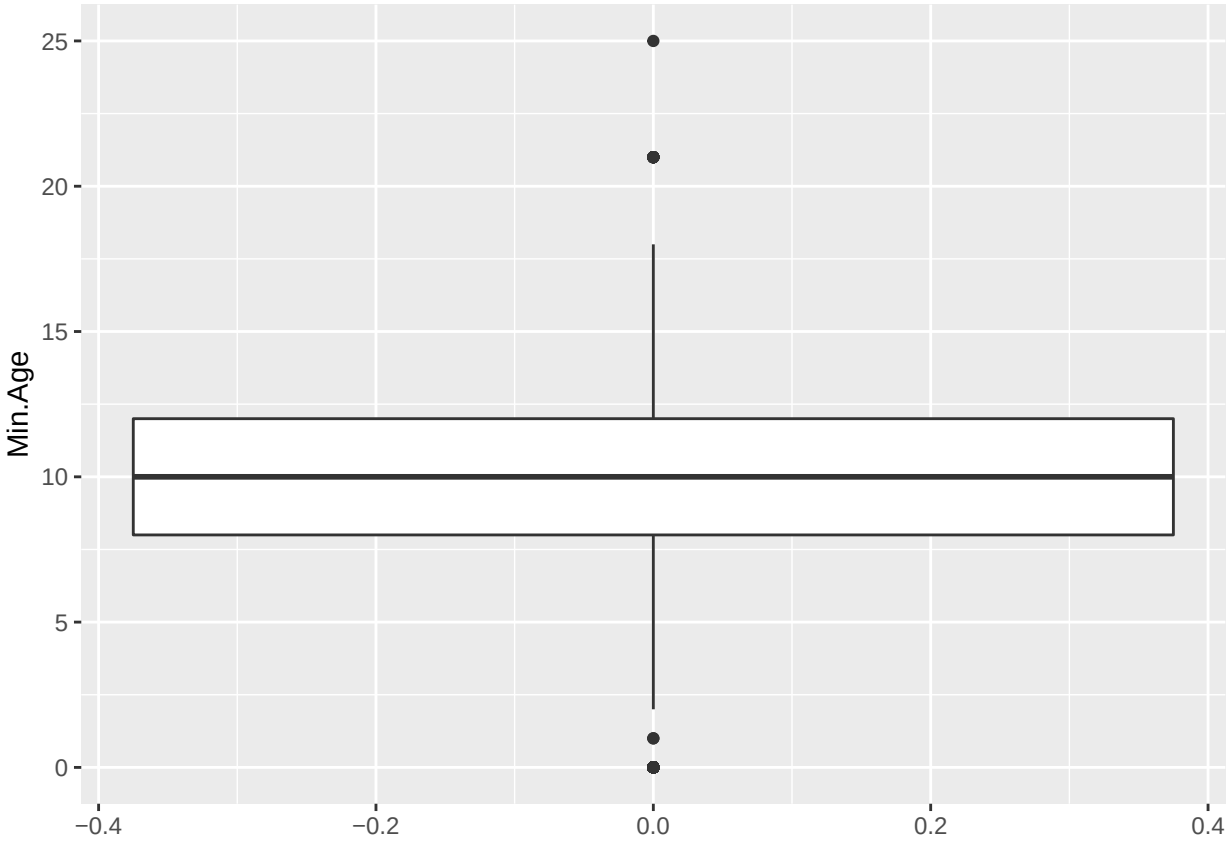

```
ggplot(dt, aes(y = Max.Players)) + geom_boxplot()
```



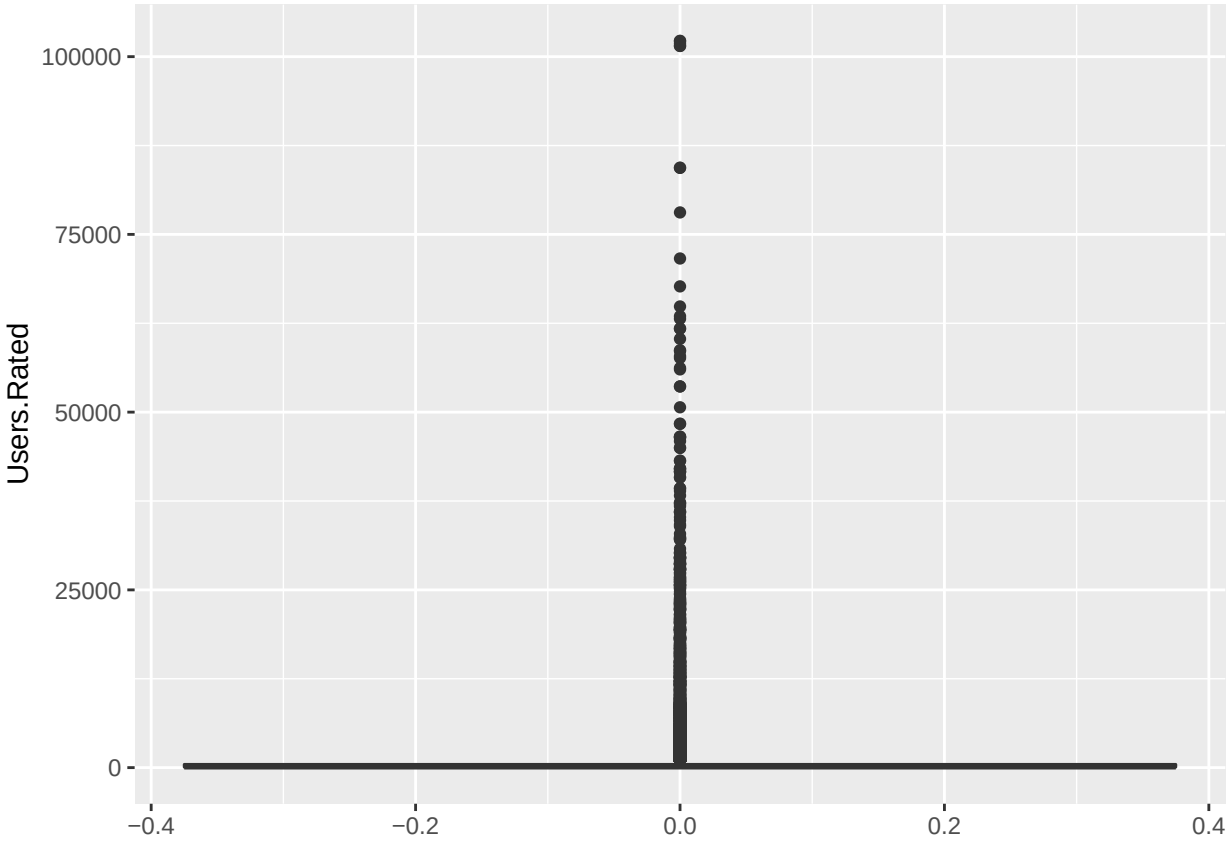

```
ggplot(dt, aes(y = Play.Time)) + geom_boxplot()
```



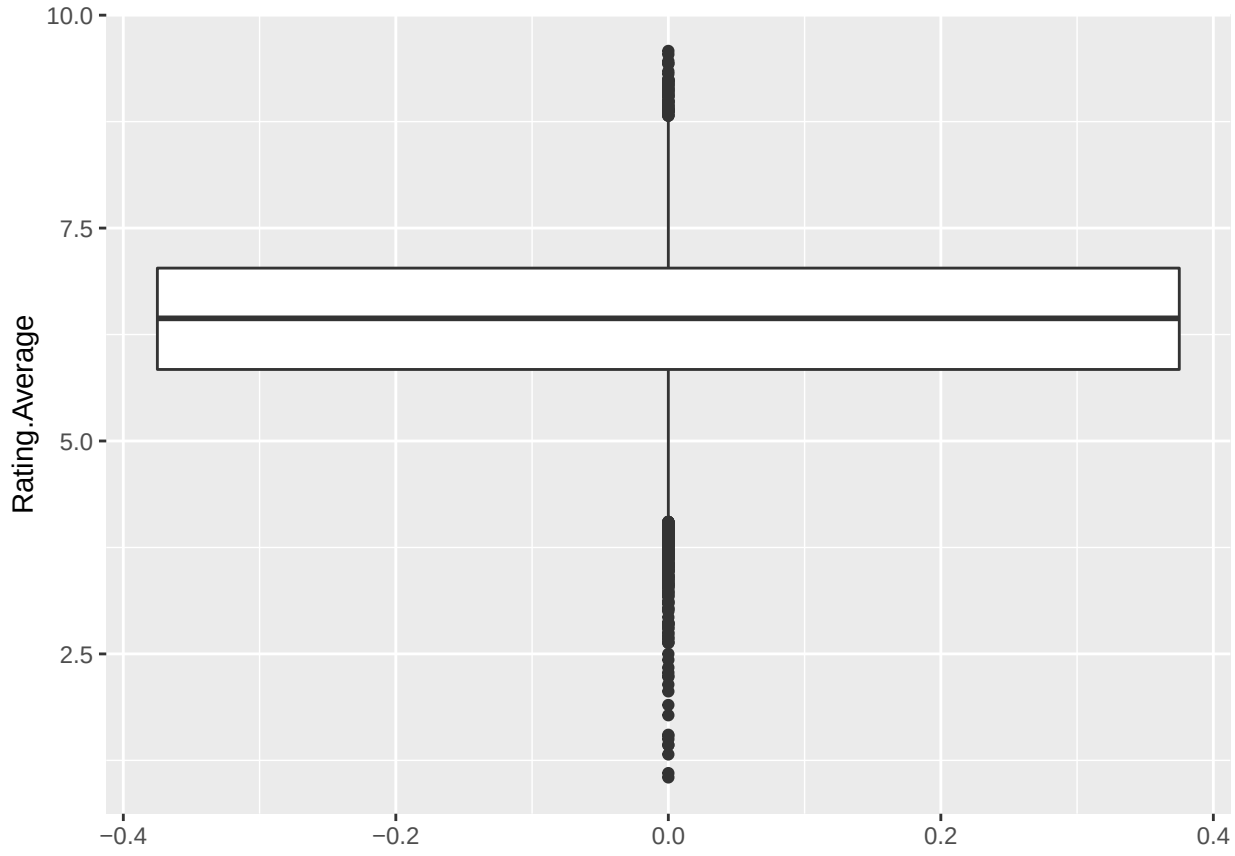

```
ggplot(dt, aes(y = Min.Age)) + geom_boxplot()
```



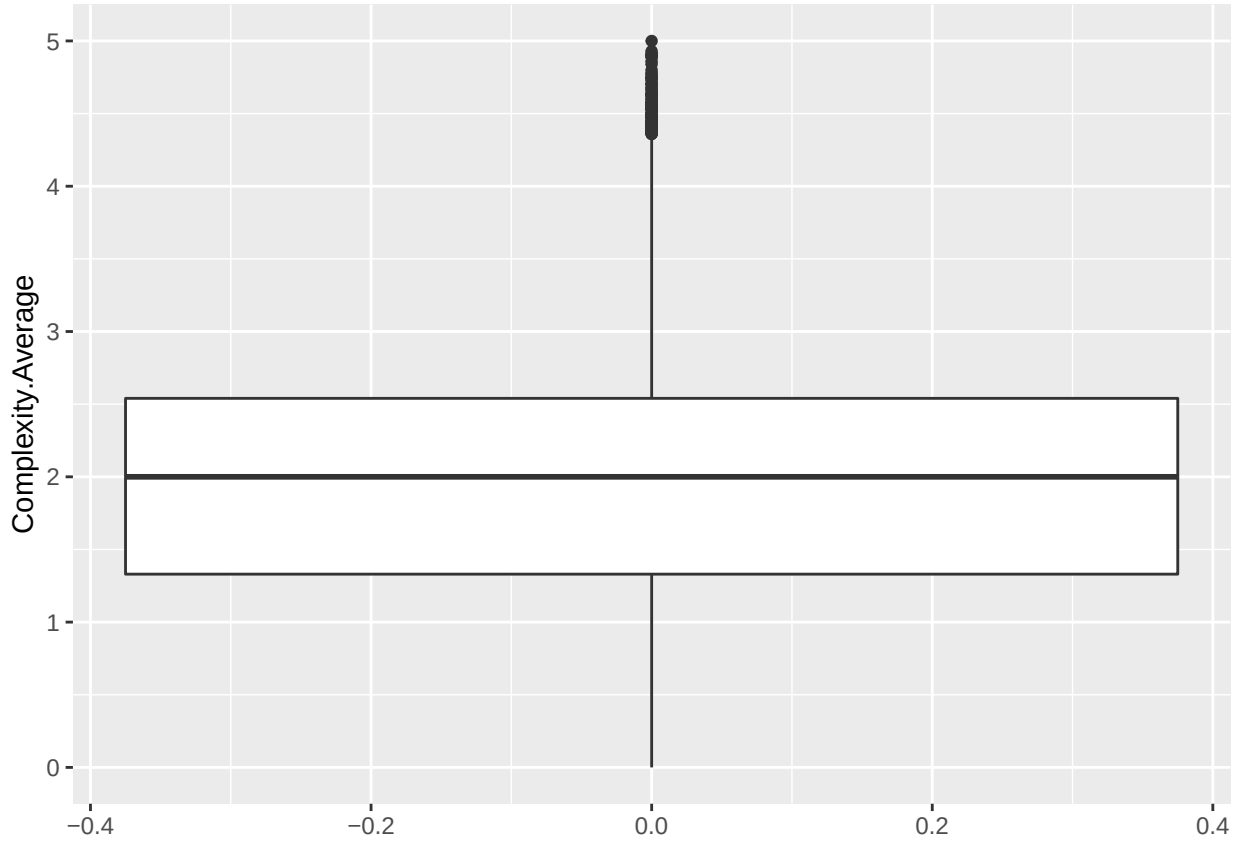

```
ggplot(dt, aes(y = Users.Rated)) + geom_boxplot()
```



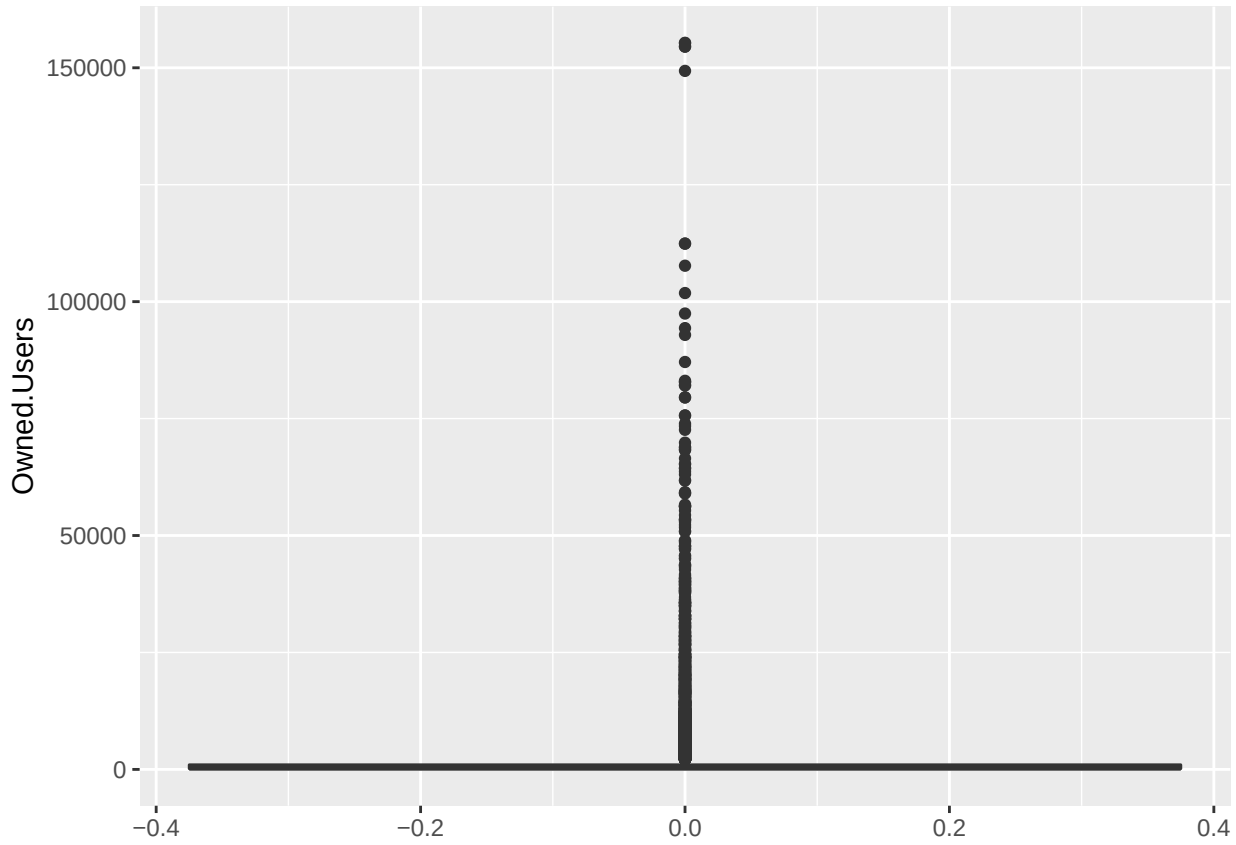

```
ggplot(dt, aes(y = Rating.Average)) + geom_boxplot()
```




```
ggplot(dt, aes(y = Complexity.Average)) + geom_boxplot()
```




```
ggplot(dt, aes(y = Owned.Users)) + geom_boxplot()
```



Como se puede apreciar en todas las gráficas se presentan registros con valores extremos, por lo que decidimos acotar estos valores. Cabe destacar, que algunas de las variables no serán acotadas ya que por lo que significan no parece tener demasiado sentido hacerlo, una de ellas es *Rating.Average*.

```
dt[Year.Published < as.integer(IQR(dt[, Year.Published])*(-1.5) +
  quantile(dt[, Year.Published], probs = 0.25)),
  Year.Published := as.integer(IQR(dt[, Year.Published])*(-1.5) +
  quantile(dt[, Year.Published], probs = 0.25))]

dt[Max.Players > IQR(dt[, Max.Players])*1.5 +
  quantile(dt[, Max.Players], probs = 0.75),
  Max.Players := IQR(dt[, Max.Players])*1.5 +
  quantile(dt[, Max.Players], probs = 0.75)]

dt[Play.Time > IQR(dt[, Play.Time])*1.5 +
  quantile(dt[, Play.Time], probs = 0.75),
  Play.Time := IQR(dt[, Play.Time])*1.5 +
  quantile(dt[, Play.Time], probs = 0.75)]

dt[Users.Rated > IQR(dt[, Users.Rated])*1.5 +
  quantile(dt[, Users.Rated], probs = 0.75),
  Users.Rated := IQR(dt[, Users.Rated])*1.5 +
  quantile(dt[, Users.Rated], probs = 0.75)]

dt[Owned.Users > IQR(dt[, Owned.Users])*1.5 +
  quantile(dt[, Owned.Users], probs = 0.75),
  Owned.Users := IQR(dt[, Owned.Users])*1.5 +
```

```
  quantile(dt[, Owned.Users], probs = 0.75)]
```

Análisis de los datos

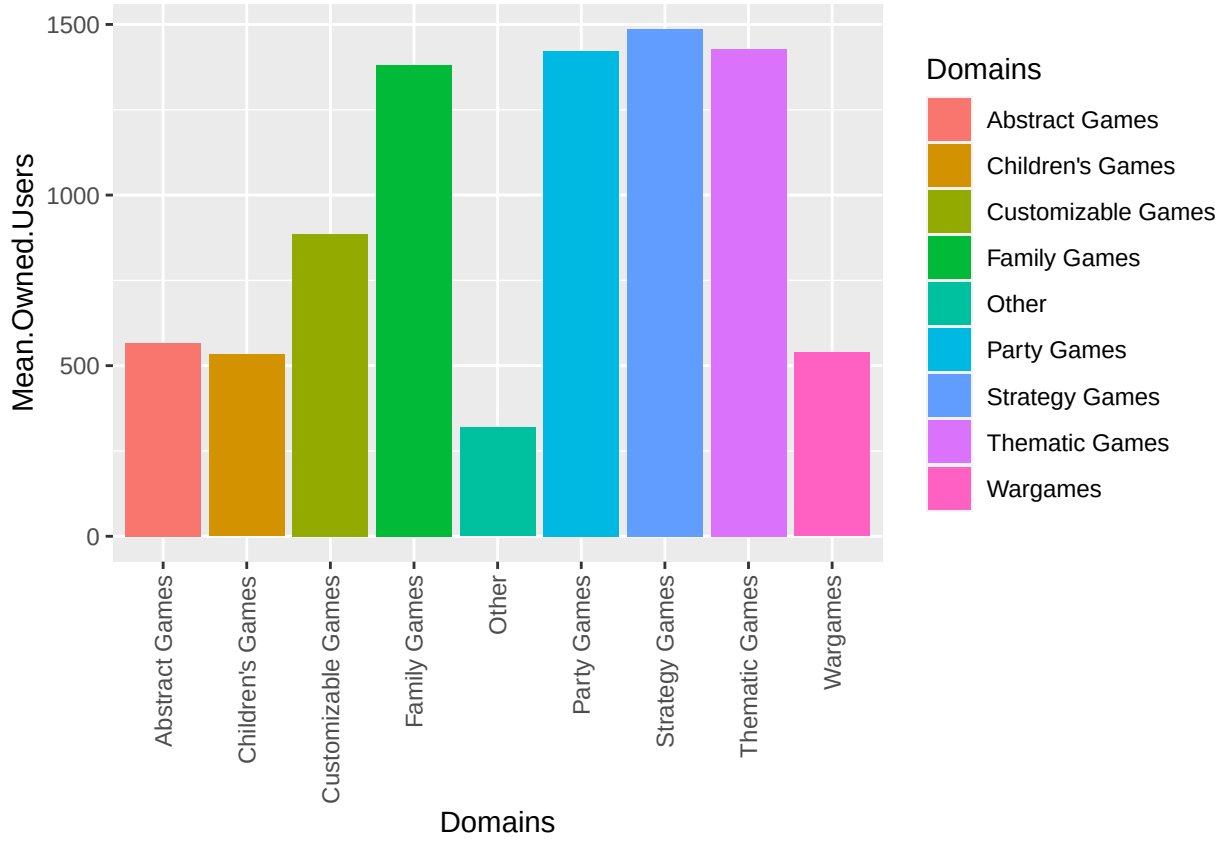
Comparación de las variables

Ahora mostraremos algunas gráficas las cuales pueden ayudarnos a entender mejor nuestros datos, llegando incluso a dar respuesta a alguna de las preguntas que hemos presentado al principio de la práctica.

Comenzaremos mostrando una gráfica que compara

```
dtPlot <- dt[, .(Domains, Owned.Users)] %>%
  group_by(Domains) %>%
  summarise(Mean.Owned.Users = mean(Owned.Users))

ggplot(data=dtPlot, aes(x=Domains, y=Mean.Owned.Users, fill = Domains)) +
    geom_bar(stat="identity", position="stack") +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1))
```



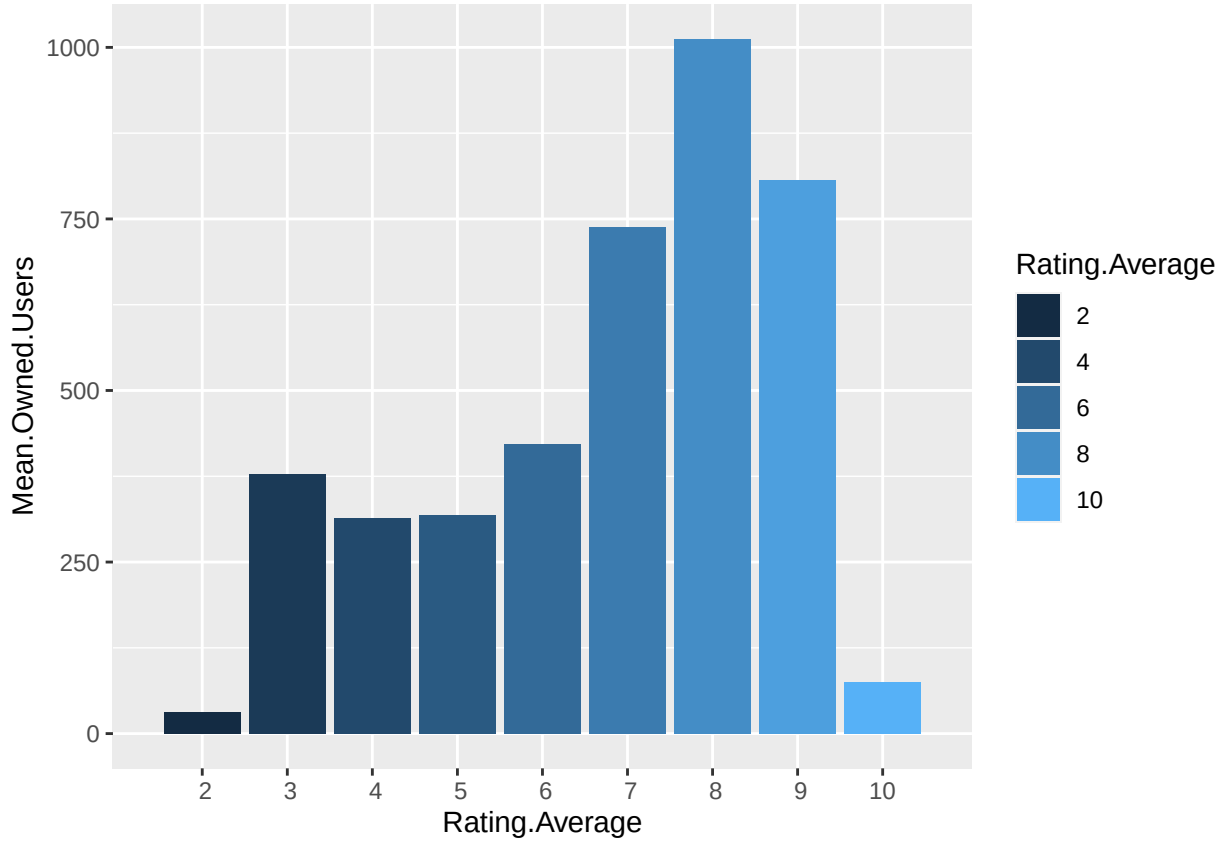
Como se puede apreciar algunos géneros son más populares que otros, estos son los juegos familiares, los de fiesta, los de estrategía y los temáticos, con este resultado puede darse respuesta a la primera de las preguntas que hemos planteado, es decir, existen algunos tipos de juegos que se venden mejor que otros, y estos son los mencionados anteriormente.

Ahora la intención es comparar la variable de la media de puntuación con el total de ventas, de esta manera podremos ver si la puntuación influye en el número de ventas.

```
dtPlot <- dt %>% data.table() %>% mutate(Rating.Average.2 = ceiling(Rating.Average))

dtPlot <- dtPlot[, .(Rating.Average.2, Owned.Users)] %>%
  group_by(Rating.Average.2) %>%
  summarise(Mean.Owned.Users = mean(Owned.Users))

ggplot(data=dtPlot, aes(x=Rating.Average.2, y=Mean.Owned.Users, fill = Rating.Average.2)) +
    geom_bar(stat="identity", position="stack") +
  guides(fill=guide_legend(title="Rating.Average")) +
  scale_x_discrete(name ="Rating.Average", limits=c(2:10))
```

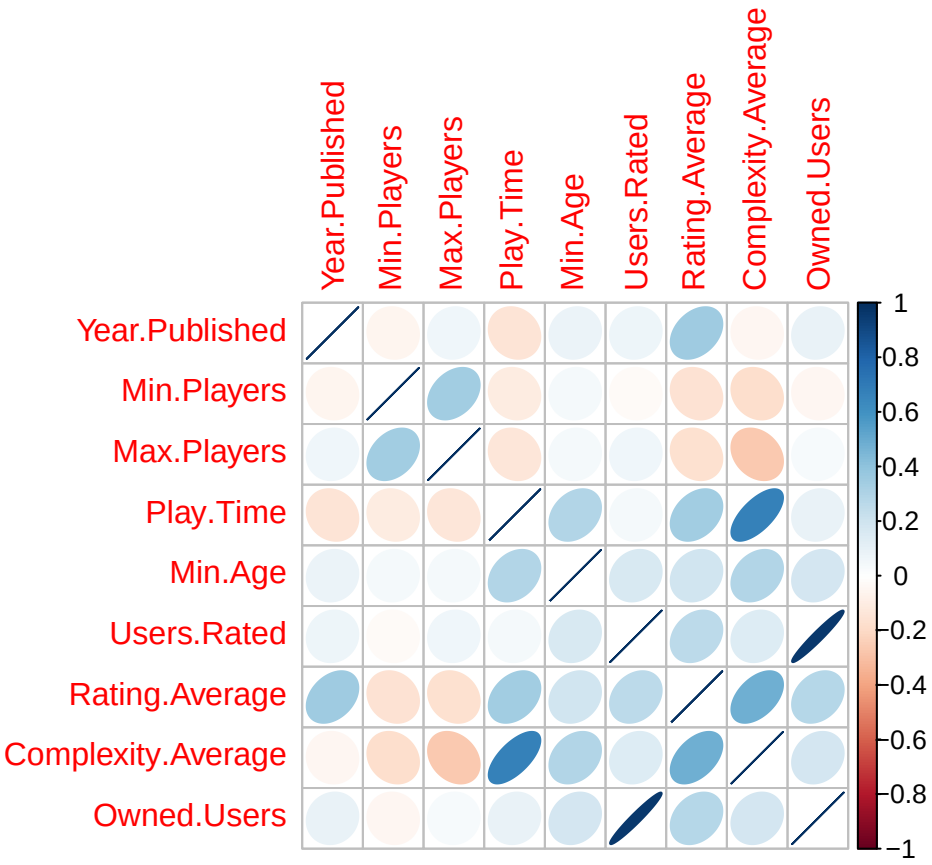


Como se puede apreciar, la puntuación de cada juego no influye en el número de ventas, ya que los juegos con una puntuación de 10 no son más vendidos que los juegos que tienen notas más bajas.

Regresión

El objetivo de este apartado es desarrollar un modelo de regresión lineal el cual pueda aproximoar la cantidad de ventas de un juego de mesa cualquiera. Antes de crear el modelo, veremos que variables estan correlacionadas entre si, ya que no nos interesa que estas formen parte de las variables que entren al modelo

```
cm <- cor(dt %>% dplyr::select(where(is.numeric)))
corrplot(cm, method = "ellipse")
```



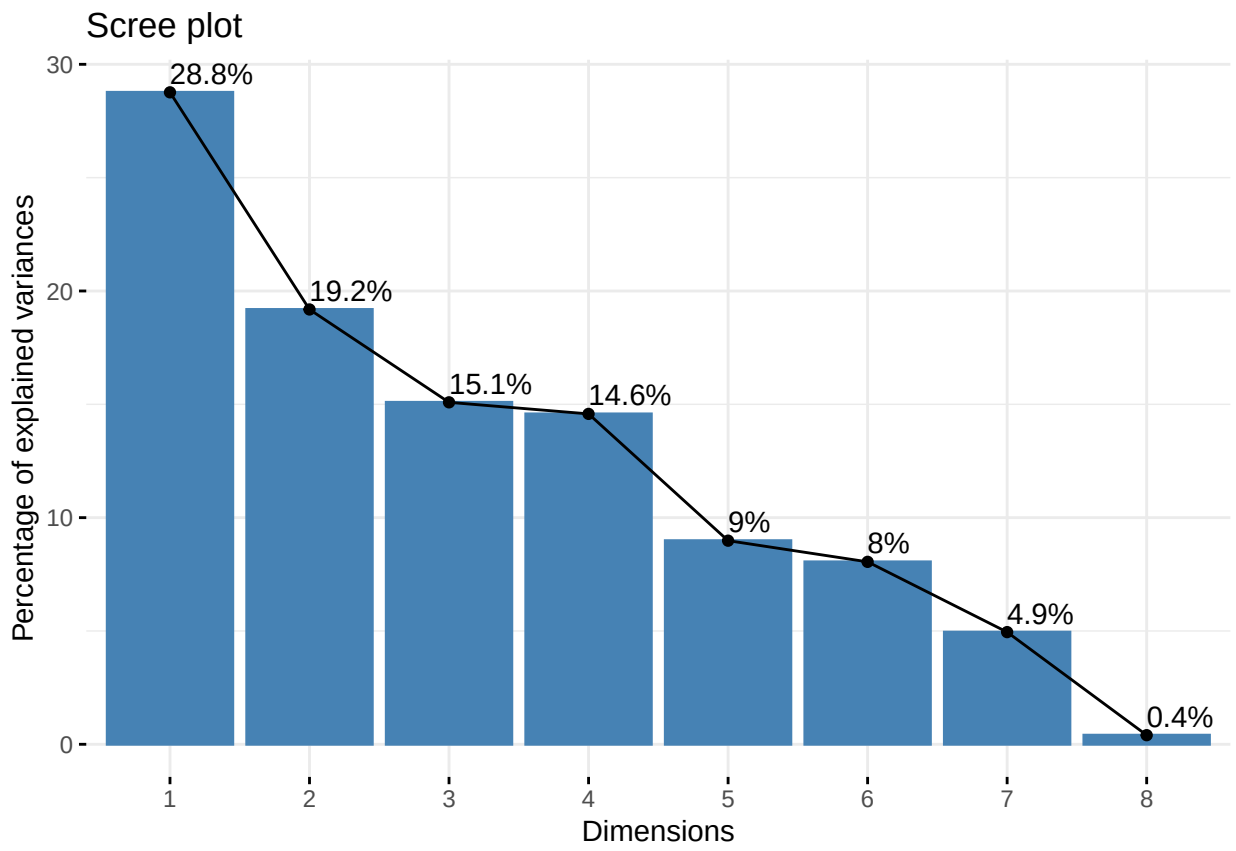
De manera general se puede apreciar que la mayoria de las variables no presentan correlación, con la excepción de *Complexity.Average*, la cual esta relacionada de manera proporcional con la variable *Play.Time* e inversamente proporcional con la variable *Max.Players*, es por eso que tomamos la decisión de excluir la variable de la complejidad de los datos que disponemos.

```
dt <- dt %>% dplyr::select(-Complexity.Average)
```

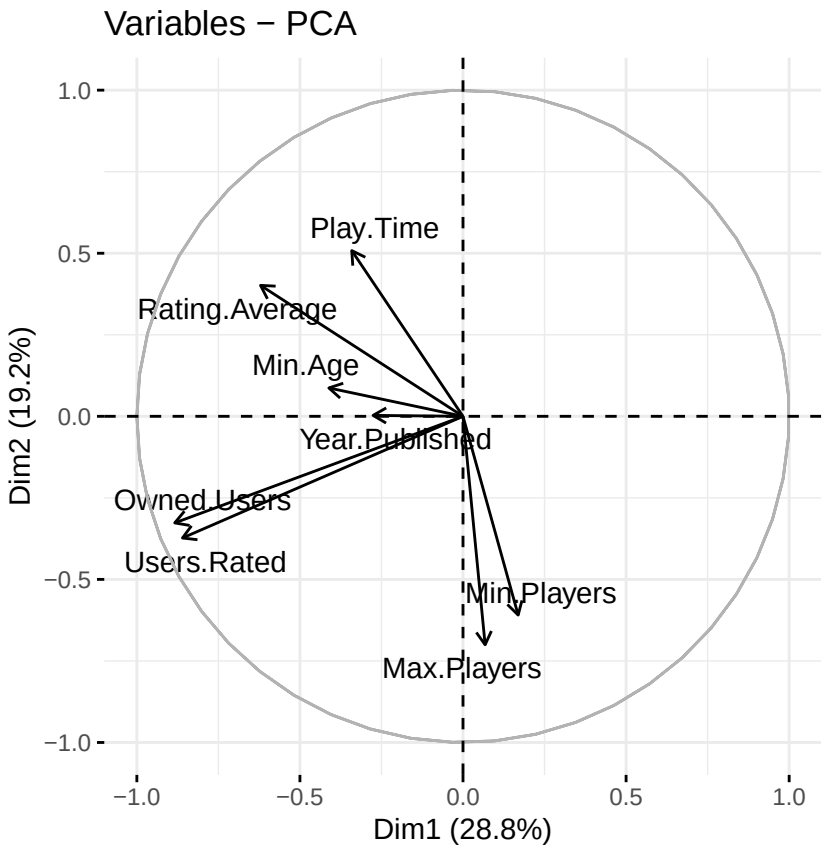
Ahora que disponemos de variables que no estan correlacionadas entre si, veremos a partir del PCA cual es la cantidad de variables que necesitamos para tener una explicabilidad lo suficientemente alta como para dar por válido el modelo

```
par(mfrow = c(1, 2))

res.pca <- prcomp(dt %>% dplyr::select(where(is.numeric)), scale = T)
fviz_eig(res.pca, addlabels = T)
```




```
fviz_pca_var(res.pca, axes =c(1,2), repel = T)
```



Podemos ver que con 6 o 7 variables obtendremos un modelo con una alta explicabilidad. Aún así, vamos a aplicar el método de selección de variables de Akaike, para ello creamos un modelo en el cual no se ha descartado ninguna variable

```
modelo <- lm(Owned.Users ~., data = dt)
summary(modelo)
```

```
##
## Call:
## lm(formula = Owned.Users ~ ., data = dt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1684.29   -73.50    -5.89    79.31  1992.27
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -3.706e+03  2.384e+02 -15.548  < 2e-16 ***
## Year.Published              1.843e+00  1.213e-01  15.198  < 2e-16 ***
## Min.Players                -1.286e+01  1.964e+00  -6.547 5.99e-11 ***
## Max.Players                -3.754e+00  7.232e-01  -5.191 2.12e-07 ***
## Play.Time                   3.813e-01  3.260e-02  11.694  < 2e-16 ***
## Min.Age                     3.974e+00  3.770e-01  10.543  < 2e-16 ***
## Users.Rated                 2.039e+00  4.865e-03 419.250  < 2e-16 ***
## Rating.Average             -1.949e+00  1.682e+00  -1.158 0.246673
## DomainsChildren's Games     4.028e+01  8.539e+00   4.717 2.41e-06 ***
## DomainsCustomizable Games   1.738e+01  1.203e+01   1.445 0.148609
```

```
## DomainsFamily Games       -2.690e+01  7.213e+00  -3.730 0.000192 ***
## DomainsOther               5.950e+01  6.250e+00   9.520  < 2e-16 ***
## DomainsParty Games         2.137e+01  9.950e+00   2.148 0.031730 *
## DomainsStrategy Games     -2.924e+01  7.437e+00  -3.932 8.45e-05 ***
## DomainsThematic Games      5.041e+01  8.232e+00   6.124 9.30e-10 ***
## DomainsWargames            1.226e+02  7.123e+00  17.214  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 182.2 on 21809 degrees of freedom
## Multiple R-squared:  0.9394, Adjusted R-squared:  0.9394
## F-statistic: 2.256e+04 on 15 and 21809 DF,  p-value: < 2.2e-16
```

```
stepAIC(modelo,direction = c("backward"))
```

```
## Start:  AIC=227227.7
## Owned.Users ~ Year.Published + Min.Players + Max.Players + Play.Time +
##     Min.Age + Users.Rated + Rating.Average + Domains
##
##                   Df  Sum of Sq        RSS    AIC
## - Rating.Average   1      44572  724329001 227227
## <none>                          724284429 227228
## - Max.Players      1     894751  725179180 227253
## - Min.Players      1    1423643  725708072 227269
## - Min.Age          1    3691605  727976034 227337
## - Play.Time        1    4541735  728826164 227362
## - Year.Published   1    7670771  731955200 227456
## - Domains          8   29509224  753793653 228083
## - Users.Rated      1 5837408610 6561693039 275324
##
## Step:  AIC=227227
## Owned.Users ~ Year.Published + Min.Players + Max.Players + Play.Time +
##     Min.Age + Users.Rated + Domains
##
##                   Df  Sum of Sq        RSS    AIC
## <none>                          724329001 227227
## - Max.Players      1     865208  725194209 227251
## - Min.Players      1    1399939  725728941 227267
## - Min.Age          1    3673163  728002164 227335
## - Play.Time        1    4600550  728929552 227363
## - Year.Published   1    8823810  733152811 227489
## - Domains          8   29466243  753795244 228081
## - Users.Rated      1 6032032454 6756361456 275960
```

```
##
## Call:
## lm(formula = Owned.Users ~ Year.Published + Min.Players + Max.Players +
##     Play.Time + Min.Age + Users.Rated + Domains, data = dt)
##
## Coefficients:
##                (Intercept)             Year.Published
##                  -3597.051                      1.783
##                Min.Players                Max.Players
##                    -12.729                     -3.675
##                  Play.Time                    Min.Age
```

```
##                      0.372                      3.963
##               Users.Rated    DomainsChildren's Games
##                      2.038                     41.461
## DomainsCustomizable Games        DomainsFamily Games
##                     17.834                    -26.334
##              DomainsOther         DomainsParty Games
##                     60.089                     21.886
##     DomainsStrategy Games      DomainsThematic Games
##                    -29.082                     50.739
##           DomainsWargames
##                    122.346
```

Este método nos devuelve la siguiente formula *Owned.Users ~ Year.Published + Min.Players + Max.Players + Play.Time + Min.Age + Users.Rated + Domains*, la cual pasará a ser la que usemos en nuestro modelo final. Por tanto, definimos nuestro nuevo modelo a partir de dicha formula

```
modeloFinal <- lm(formula = Owned.Users ~ Year.Published + Min.Players + Max.Players +
    Play.Time + Min.Age + Users.Rated + Domains, data = dt)

summary(modeloFinal)
```

```
##
## Call:
## lm(formula = Owned.Users ~ Year.Published + Min.Players + Max.Players +
##     Play.Time + Min.Age + Users.Rated + Domains, data = dt)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1678.16   -73.43    -5.92    79.21  1989.52
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -3.597e+03  2.189e+02 -16.435  < 2e-16 ***
## Year.Published              1.783e+00  1.094e-01  16.300  < 2e-16 ***
## Min.Players                -1.273e+01  1.961e+00  -6.493 8.62e-11 ***
## Max.Players                -3.675e+00  7.199e-01  -5.104 3.35e-07 ***
## Play.Time                   3.720e-01  3.161e-02  11.770  < 2e-16 ***
## Min.Age                     3.963e+00  3.768e-01  10.517  < 2e-16 ***
## Users.Rated                 2.038e+00  4.783e-03 426.179  < 2e-16 ***
## DomainsChildren's Games     4.146e+01  8.477e+00   4.891 1.01e-06 ***
## DomainsCustomizable Games   1.783e+01  1.202e+01   1.483 0.138049
## DomainsFamily Games        -2.633e+01  7.196e+00  -3.660 0.000253 ***
## DomainsOther                6.009e+01  6.230e+00   9.645  < 2e-16 ***
## DomainsParty Games          2.189e+01  9.940e+00   2.202 0.027686 *
## DomainsStrategy Games      -2.908e+01  7.436e+00  -3.911 9.22e-05 ***
## DomainsThematic Games       5.074e+01  8.227e+00   6.167 7.06e-10 ***
## DomainsWargames             1.223e+02  7.119e+00  17.185  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 182.2 on 21810 degrees of freedom
## Multiple R-squared:  0.9394, Adjusted R-squared:  0.9394
## F-statistic: 2.417e+04 on 14 and 21810 DF,  p-value: < 2.2e-16
```

En este modelo se ha decidido prescindir de la variable *Rating.Average*, y además para la variable *Domains* se han construido variables dummy las cuales permiten considerar todas las categorías que esta variable tiene.

Por otro lado, se puede apreciar que obtenemos un R-squared bastante próximo a 1, siendo este de 0.9394, lo que es un buen indicativo de que el modelo funcionará de manera adecuada.

Normalidad y homocedasticidad

Ahora nos tocará verificar que nuestro modelo cumple las condiciones de normalidad y de igualdad de varianzaas. Para ello usaremos distintos test estadísticos.

```
lillie.test(modeloFinal$residuals)
```

```
##
##  Lilliefors (Kolmogorov-Smirnov) normality test
##
## data:  modeloFinal$residuals
## D = 0.11475, p-value < 2.2e-16
```

```
ad.test(modeloFinal$residuals)
```

```
##
##  Anderson-Darling normality test
##
## data:  modeloFinal$residuals
## A = 638.65, p-value < 2.2e-16
```

```
pearson.test(modeloFinal$residuals)
```

```
##
##  Pearson chi-square normality test
##
## data:  modeloFinal$residuals
## P = 6358, p-value < 2.2e-16
```

```
cvm.test(modeloFinal$residuals)
```

```
## Warning in cvm.test(modeloFinal$residuals): p-value is smaller than 7.37e-10,
## cannot be computed more accurately
```

```
##
##  Cramer-von Mises normality test
##
## data:  modeloFinal$residuals
## W = 109.37, p-value = 7.37e-10
```

Comprobemos hora la homocedasticidad mediante los test de Levene y el test de Fligner-Killeen

```
car::leveneTest(y = dt$Owned.Users, group = dt$Domains, center = "median")
```

```
## Levene's Test for Homogeneity of Variance (center = "median")
##          Df F value    Pr(>F)
## group     8  817.14 < 2.2e-16 ***
##       21816
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

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

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Owned.Users by Rating.Average
```

```
## Fligner-Killeen:med chi-squared = 3682.4, df = 620, p-value < 2.2e-16
```

```
fligner.test(Owned.Users ~ Min.Players, data = dt)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Owned.Users by Min.Players
## Fligner-Killeen:med chi-squared = 105.85, df = 10, p-value < 2.2e-16
```

```
fligner.test(Owned.Users ~ Max.Players, data = dt)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Owned.Users by Max.Players
## Fligner-Killeen:med chi-squared = 595.28, df = 9, p-value < 2.2e-16
```

```
fligner.test(Owned.Users ~ Play.Time, data = dt)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  Owned.Users by Play.Time
## Fligner-Killeen:med chi-squared = 1249.1, df = 53, p-value < 2.2e-16
```

Lamentablemente nuestro modelo no cumple ni la condición de normalidad ni la de homocedasticidad, por lo que se puede considerar aplicar las transformaciones de Box-Cox y construir nuevamente el modelo, otra opción sería, no realizar ninguna transformación pero hacer uso de otro tipo de modelo.

Conslusión

Algunas de las cuestiones de las que se plantearon inicialmente y que quedan por responder son las siguientes

¿La duración de las partidas afecta a las ventas?

Esta pregunta se podría responder mostrando un gráfico con los datos que tenemos, pero también se puede hacer teniendo en cuenta el modelo que hemos construido, en él, se puede apreciar que la variable *Play.Time* entra al modelo y tiene un peso de 0.372, esto se puede interpretar como que efectivamente, el tiempo de las partidas influye en las ventas de un juego mesa de la forma que a mayor tiempo medio de las partidas mayores serán las ventas.

¿Es posible estimar las ventas que tendrá un juego de mesa a partir de los datos que disponemos?

Tal y como hemos hecho anteriormente, solo habrá que tratar los datos que tenemos y desarrollar un modelo el cual estime el número de ventas. En esta práctica hemos optado por una regresión lineal, pero se podrían haber aplicado técnicas más complejas para el desarrollo del modelo.

Otras cuestiones

También se pueden plantear cuestiones más a largo plazo, como podrían ser preguntas sobre como van a evolucionar ciertas variables a lo largo del tiempo, es decir, no solo predecir si las ventas aumentarán en los próximos años, si no que también ser capaces de identificar las tendencias de los tipos de juegos que habrá en los años venideros, es por este motivo que es de gran importancia tener un visión global de negocio.

Contribuciones

Contribuciones	Firma
Investigación previa	Jorge Ramón Díaz Suarez, Víctor Fernández Moreno
Redacción de las respuestas	Jorge Ramón Díaz Suarez, Víctor Fernández Moreno
Desarrollo del código	Jorge Ramón Díaz Suarez, Víctor Fernández Moreno
Participación en el vídeo	Jorge Ramón Díaz Suarez, Víctor Fernández Moreno