PRACTICA 2: LIMPIEZA Y VALIDACIÓN DE LOS DATOS

28 de mayo 2021

---

1. Descripción del dataset.

El dataset que hemos selecionado contiene datos de los 100 juegos de Google Play Store mejor valorados. Las variables que recoge el dataset son:

- Rank: Calsificación de una categoría particular.
- Title: Nombre del juego.
- Total ranting: Número total de calificaciones.
- Installs: Número de intalaciones aproximado.
- Average rating: Promedio de estrellas.
- growth (30 days): Porcentaje de crecimiento en 30 días.
- growth (60 days): Porcentaje de crecimiento en 60 días.
- price: Precio en Dolares.
- category: Cantegoria del juego.
- X5.star.ratings: Número de calificaciones de 5 estrellas.
- X4.star.ratings: Número de calificaciones de 4 estrellas.
- X3.star.ratings: Número de calificaciones de 3 estrellas.
- X2.star.ratings: Número de calificaciones de 2 estrellas.
- X1.star.ratings: Número de calificaciones de 1 estrellas.
- paid: Toma valor verdadero si se pago y falso si no se hizo.

¿Por qué es importante y qué pregunta/problema pretende responder?

---

2. Integración y selección de los datos de interés a analizar.

2.1. Integración de los datos.

En primer lugar, frealizamos la carga del fichero que contiene los datos para nuestro análisis en formato csv, el cual está delimitado por comas y los decismales " · ". Obtenemos como resultado de la llamada a la función read.csv() será un objeto data.frame.

```
# read data
games <- read.csv("/Users/moreybagarciacedres/Downloads/android-games.csv", header=TRUE, sep=",", na.st
n.var <- names(games)

#View(games)
```

Mostramos las primeras lineas del dataset, así como su encabezado.

```
head(games)
```

```
##   rank                                       title total.ratings installs
## 1    1                Garena Free Fire - The Cobra      80678661  500.0 M
## 2    2                 PUBG MOBILE: Graffiti Prank      35971961  100.0 M
## 3    3                   Mobile Legends: Bang Bang      25836869  100.0 M
## 4    4                                Brawl Stars      17181659  100.0 M
## 5    5 Sniper 3D: Fun Free Online FPS Shooting Game      14237554  100.0 M
## 6    6                              Shadow Fight 2      14048931  100.0 M
##   average.rating growth..30.days. growth..60.days. price    category
## 1           4.33              2.9              7.9     0 GAME ACTION
## 2           4.24              2.0              3.1     0 GAME ACTION
## 3           4.08              1.6              3.3     0 GAME ACTION
## 4           4.27              4.1              6.6     0 GAME ACTION
## 5           4.33              0.8              1.8     0 GAME ACTION
## 6           4.57              0.6              1.5     0 GAME ACTION
##   X5.star.ratings X4.star.ratings X3.star.ratings X2.star.ratings
## 1        61935712         4478738         2795172         1814999
## 2        26670566         2109631         1352610          893674
## 3        17850942         1796761         1066095          725429
## 4        12493668         1474319          741410          383478
## 5         9657878         2124544         1034025          375159
## 6        11532143          961926          448184          217044
##   X1.star.ratings  paid
## 1         9654037 False
## 2         4945478 False
## 3         4397640 False
## 4         2088781 False
## 5         1045945 False
## 6          889631 False
```

Una vez cargados los datos comprobamos que nuestro fichero contiene 1730 registros y 15 variables.

Las variables son de tipo rank, title, total.ratings, installs, average.rating, growth..30.days., growth..60.days., price, category, X5.star.ratings, X4.star.ratings, X3.star.ratings, X2.star.ratings, X1.star.ratings, paid.

Como podemos en el resumen de los estadisticos descriptivos de las distintas variables no hay ningún campo no informado.

```
summary(games)
```

```
##       rank           title           total.ratings        installs
##  Min.   :  1.00   Length:1730        Min.   :   38238   Length:1730
##  1st Qu.: 25.00   Class :character   1st Qu.:  187999   Class :character
##  Median : 51.00   Mode  :character   Median :  457675   Mode  :character
##  Mean   : 50.48                      Mean   : 1101182
##  3rd Qu.: 75.75                      3rd Qu.:  944334
##  Max.   :100.00                      Max.   :80678661
##  average.rating  growth..30.days.   growth..60.days.      price
##  Min.   :3.090   Min.   :     0.0   Min.   :  0.000   Min.   :0.00000
##  1st Qu.:4.180   1st Qu.:     0.1   1st Qu.:  0.300   1st Qu.:0.00000
##  Median :4.330   Median :     0.5   Median :  1.000   Median :0.00000
##  Mean   :4.313   Mean   :   193.2   Mean   :  3.969   Mean   :0.01297
##  3rd Qu.:4.490   3rd Qu.:     1.6   3rd Qu.:  3.300   3rd Qu.:0.00000
##  Max.   :4.910   Max.   :140394.4   Max.   :605.100   Max.   :7.49000
##    category         X5.star.ratings    X4.star.ratings   X3.star.ratings
```

```
##  Length:1730        Min.   :   21898   Min.   :   2441   Min.   :    707
##  Class :character   1st Qu.:  135829   1st Qu.:  21802   1st Qu.:  10278
##  Mode  :character   Median :  310944   Median :  54644   Median :  26658
##                     Mean   :  788384   Mean   : 121647   Mean   :  59550
##                     3rd Qu.:  651131   3rd Qu.: 109565   3rd Qu.:  55818
##                     Max.   :61935712   Max.   :5397273   Max.   :2795172
##  X2.star.ratings   X1.star.ratings       paid
##  Min.   :    288   Min.   :    527   Length:1730
##  1st Qu.:   4530   1st Qu.:  13561   Class :character
##  Median :  11330   Median :  35694   Mode  :character
##  Mean   :  27962   Mean   : 103636
##  3rd Qu.:  25266   3rd Qu.:  86326
##  Max.   :1814999   Max.   :9654037
```

2.2. Tipo variables.

Comprobamos de que tipo es cada variable.

```
#read data
res <- sapply(games,  class)
kable(data.frame(variables=names(res),clase=as.vector(res)))
```

variables	clase
rank	integer
title	character
total.ratings	integer
installs	character
average.rating	numeric
growth..30.days.	numeric
growth..60.days.	numeric
price	numeric
category	character
X5.star.ratings	integer
X4.star.ratings	integer
X3.star.ratings	integer
X2.star.ratings	integer
X1.star.ratings	integer
paid	character

A continuación analizamos en mayor profundidad los distintos valores que toman las variables categóricas.

```
title <-unique(games$title)
head(title)
```

```
## [1] "Garena Free Fire - The Cobra"
## [2] "PUBG MOBILE: Graffiti Prank"
## [3] "Mobile Legends: Bang Bang"
## [4] "Brawl Stars"
## [5] "Sniper 3D: Fun Free Online FPS Shooting Game"
## [6] "Shadow Fight 2"
```

```
unique(games$category)
```

```
##  [1] "GAME ACTION"       "GAME ADVENTURE"    "GAME ARCADE"
##  [4] "GAME BOARD"        "GAME CARD"         "GAME CASINO"
```

```
##  [7] "GAME CASUAL"       "GAME EDUCATIONAL"  "GAME MUSIC"
## [10] "GAME PUZZLE"       "GAME RACING"       "GAME ROLE PLAYING"
## [13] "GAME SIMULATION"   "GAME SPORTS"       "GAME STRATEGY"
## [16] "GAME TRIVIA"       "GAME WORD"
```

```
unique(games$installs)
```

```
## [1] "500.0 M"  "100.0 M"  "50.0 M"   "10.0 M"   "5.0 M"    "1.0 M"    "1000.0 M"
## [8] "500.0 k"  "100.0 k"
```

```
unique(games$paid)
```

```
## [1] "False" "True"
```

Observamos que la variable title toma un valor distinto para cada registro. Más adelante trataresmos el resto de variables que sean necesarias para nuestro análisis.

2.3. Selección de variables.

Para nuestro análisis vamos a eliminar del dataset los campos calculados, en nuetro caso son la media de estrellas que es la variable **average.rating, paid y total.ratings**. Por otro lado, para nuestro analisis es suficiente con uno de los porcentajes de crecimiento por tanto eliminamos tambien la variable **growth..60.days**.

```
games <-games[,c(-3,-5,-7,-15)] # Elimienamos las variables en cuestión.

head(games)
```

```
##   rank                                          title installs growth..30.days.
## 1    1                 Garena Free Fire - The Cobra  500.0 M              2.9
## 2    2                  PUBG MOBILE: Graffiti Prank  100.0 M              2.0
## 3    3                    Mobile Legends: Bang Bang  100.0 M              1.6
## 4    4                                  Brawl Stars  100.0 M              4.1
## 5    5 Sniper 3D: Fun Free Online FPS Shooting Game  100.0 M              0.8
## 6    6                               Shadow Fight 2  100.0 M              0.6
##   price    category X5.star.ratings X4.star.ratings X3.star.ratings
## 1     0 GAME ACTION        61935712         4478738         2795172
## 2     0 GAME ACTION        26670566         2109631         1352610
## 3     0 GAME ACTION        17850942         1796761         1066095
## 4     0 GAME ACTION        12493668         1474319          741410
## 5     0 GAME ACTION         9657878         2124544         1034025
## 6     0 GAME ACTION        11532143          961926          448184
##   X2.star.ratings X1.star.ratings
## 1         1814999         9654037
## 2          893674         4945478
## 3          725429         4397640
## 4          383478         2088781
## 5          375159         1045945
## 6          217044          889631
```

Como resusltado obtenemos un dataset que contiene 1730 registros y 11 variables.

Las variables son de tipo rank, title, total.ratings, installs, average.rating, growth..30.days., growth..60.days., price, category, X5.star.ratings, X4.star.ratings, X3.star.ratings, X2.star.ratings, X1.star.ratings, paid.

---

3. Limpieza de los datos.

Con el archivo de datos obtenido del proceso anterior, vemos la necesidad, para seguir con nuestro análisis, de factorizar la variable **Category** y reconvertir a numérica la variable **Installs**.

A continuación, damos un valor numérico del 1 al 17 a los valores que toma la variable **Category**.

```
levels <- c(unique(games$category))
games$category_num <- match(games$category, levels)
games$category_factor = factor(games$category, levels = levels)
games$category_num2 <- as.integer(games$category_factor)

head(games)
```

```
##   rank                                      title installs growth..30.days.
## 1    1                Garena Free Fire - The Cobra  500.0 M              2.9
## 2    2                 PUBG MOBILE: Graffiti Prank  100.0 M              2.0
## 3    3                   Mobile Legends: Bang Bang  100.0 M              1.6
## 4    4                                Brawl Stars  100.0 M              4.1
## 5    5 Sniper 3D: Fun Free Online FPS Shooting Game  100.0 M              0.8
## 6    6                             Shadow Fight 2  100.0 M              0.6
##   price    category X5.star.ratings X4.star.ratings X3.star.ratings
## 1     0 GAME ACTION        61935712         4478738         2795172
## 2     0 GAME ACTION        26670566         2109631         1352610
## 3     0 GAME ACTION        17850942         1796761         1066095
## 4     0 GAME ACTION        12493668         1474319          741410
## 5     0 GAME ACTION         9657878         2124544         1034025
## 6     0 GAME ACTION        11532143          961926          448184
##   X2.star.ratings X1.star.ratings category_num category_factor category_num2
## 1         1814999         9654037            1     GAME ACTION             1
## 2          893674         4945478            1     GAME ACTION             1
## 3          725429         4397640            1     GAME ACTION             1
## 4          383478         2088781            1     GAME ACTION             1
## 5          375159         1045945            1     GAME ACTION             1
## 6          217044          889631            1     GAME ACTION             1
```

En el caso de la variable **Install** hemos supuesto que M Megabytes y que k son kilobytes, por lo que hemos multiplicado la M por 10^6 para pasalo a bytes y k por 10^3.

```
converter<-function(valueToConvert) {
   intValue <- 0
   intValue <- strtoi(substr(valueToConvert, 1, regexpr('\\.', valueToConvert)-1))
    if (grepl("M", valueToConvert, ignore.case = TRUE)) {
        intValue <- intValue * (10**6)
    }
    if (grepl("K", valueToConvert, ignore.case = TRUE)) {
       intValue <- intValue * (10**3)
    }
   intValue
}

games$int_installs <- sapply(games$installs, FUN=converter)

head(games)
```

```
##   rank                                       title installs growth..30.days.
## 1    1                 Garena Free Fire - The Cobra  500.0 M              2.9
## 2    2                  PUBG MOBILE: Graffiti Prank  100.0 M              2.0
## 3    3                    Mobile Legends: Bang Bang  100.0 M              1.6
## 4    4                                  Brawl Stars  100.0 M              4.1
## 5    5 Sniper 3D: Fun Free Online FPS Shooting Game  100.0 M              0.8
## 6    6                               Shadow Fight 2  100.0 M              0.6
##   price    category X5.star.ratings X4.star.ratings X3.star.ratings
## 1     0 GAME ACTION        61935712         4478738         2795172
## 2     0 GAME ACTION        26670566         2109631         1352610
## 3     0 GAME ACTION        17850942         1796761         1066095
## 4     0 GAME ACTION        12493668         1474319          741410
## 5     0 GAME ACTION         9657878         2124544         1034025
## 6     0 GAME ACTION        11532143          961926          448184
##   X2.star.ratings X1.star.ratings category_num category_factor category_num2
## 1         1814999         9654037            1     GAME ACTION             1
## 2          893674         4945478            1     GAME ACTION             1
## 3          725429         4397640            1     GAME ACTION             1
## 4          383478         2088781            1     GAME ACTION             1
## 5          375159         1045945            1     GAME ACTION             1
## 6          217044          889631            1     GAME ACTION             1
##   int_installs
## 1        5e+08
## 2        1e+08
## 3        1e+08
## 4        1e+08
## 5        1e+08
## 6        1e+08
```

```
games <-games[,c(-3,-6,-12,-13)] # Elimienamos las variables en variables que sobran
```

Finalmente, tras tratar las variables mal informadas nos queda nuestro dataset final sobre el que vamos a plicar la limpieza de datos y posteriormente el análisis.

```
#read data
res <- sapply(games,  class)
kable(data.frame(variables=names(res),clase=as.vector(res)))
```

variables	clase
rank	integer
title	character
growth..30.days.	numeric
price	numeric
X5.star.ratings	integer
X4.star.ratings	integer
X3.star.ratings	integer
X2.star.ratings	integer
X1.star.ratings	integer
category_num2	integer
int_installs	numeric

Como podemos ver solo nos hemos quedado con una varible categoricas **Title**y el resto las hemos reconvertido en númericas.

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

Como se puede comprobar en el resumen de los estadisticos descriptivos de las distintas variables, no hay ningún valor nulo.

```
summary(games)
```

```
##       rank           title           growth..30.days.       price
##  Min.   :  1.00   Length:1730        Min.   :     0.0   Min.   :0.00000
##  1st Qu.: 25.00   Class :character   1st Qu.:     0.1   1st Qu.:0.00000
##  Median : 51.00   Mode  :character   Median :     0.5   Median :0.00000
##  Mean   : 50.48                      Mean   :   193.2   Mean   :0.01297
##  3rd Qu.: 75.75                      3rd Qu.:     1.6   3rd Qu.:0.00000
##  Max.   :100.00                      Max.   :140394.4   Max.   :7.49000
##  X5.star.ratings    X4.star.ratings   X3.star.ratings   X2.star.ratings
##  Min.   :   21898   Min.   :   2441   Min.   :    707   Min.   :    288
##  1st Qu.:  135829   1st Qu.:  21802   1st Qu.:  10278   1st Qu.:   4530
##  Median :  310944   Median :  54644   Median :  26658   Median :  11330
##  Mean   :  788384   Mean   : 121647   Mean   :  59550   Mean   :  27962
##  3rd Qu.:  651131   3rd Qu.: 109565   3rd Qu.:  55818   3rd Qu.:  25266
##  Max.   :61935712   Max.   :5397273   Max.   :2795172   Max.   :1814999
##  X1.star.ratings   category_num2     int_installs
##  Min.   :    527   Min.   : 1.000   Min.   :1.000e+05
##  1st Qu.:  13561   1st Qu.: 5.000   1st Qu.:5.000e+06
##  Median :  35694   Median : 9.000   Median :1.000e+07
##  Mean   : 103636   Mean   : 8.975   Mean   :2.889e+07
##  3rd Qu.:  86326   3rd Qu.:13.000   3rd Qu.:5.000e+07
##  Max.   :9654037   Max.   :17.000   Max.   :1.000e+09
```

Con la función summary vemos si existen valores nulos y cual es el valor maximo y minimo que toma cada variable. La variable price toma valor cero pero es de interes para el análisis ya que significa que el usuario no ha pagando nada. Por otro lado, la variable growth..30.days. también toma valor 0 en el caso de no haber crecimiento.

3.2. Identificación y tratamiento de valores extremos.

Los valores extremos o outliers son aquellos que parecen no ser congruentes sin los comparamos con el resto de los datos. Para identificarlos, podemos hacer uso de dos vías: (1) representar un diagrama de caja por cada variable y ver qué valores distan mucho del rango intercuartílico (la caja) o (2) utilizar la función boxplots.stats() de R, la cual se emplea a continuación. Así, se mostrarán sólo los valores atípicos para aquellas variables que los contienen:

```
boxplot.stats(games$rank)$out
```

```
## integer(0)
```

```
boxplot.stats(games$int_installs)$out
```

```
##  [1] 5e+08 5e+08 1e+09 5e+08 5e+08 1e+09 5e+08 5e+08 5e+08 5e+08 5e+08 5e+08
```

```
out_growth <- boxplot.stats(games$growth..30.days.)$out
head(out_growth)
```

```
## [1]      4.1      6.3      6.1   4560.2   1164.8 139410.4
```

```
boxplot.stats(games$price)$out
```

```
## [1] 0.99 4.99 7.49 1.99 2.99 1.99 1.99
```

```
out_X5 <- boxplot.stats(games$X5.star.ratings)$out
head(out_X5)
```

```
## [1] 61935712 26670566 17850942 12493668  9657878 11532143
```

```
out_X4 <- boxplot.stats(games$X4.star.ratings)$out
head(out_X4)
```

```
## [1] 4478738 2109631 1796761 1474319 2124544  961926
```

```
out_X3 <- boxplot.stats(games$X3.star.ratings)$out
head(out_X3)
```

```
## [1] 2795172 1352610 1066095  741410 1034025  448184
```

```
out_X2 <- boxplot.stats(games$X2.star.ratings)$out
head(out_X2)
```

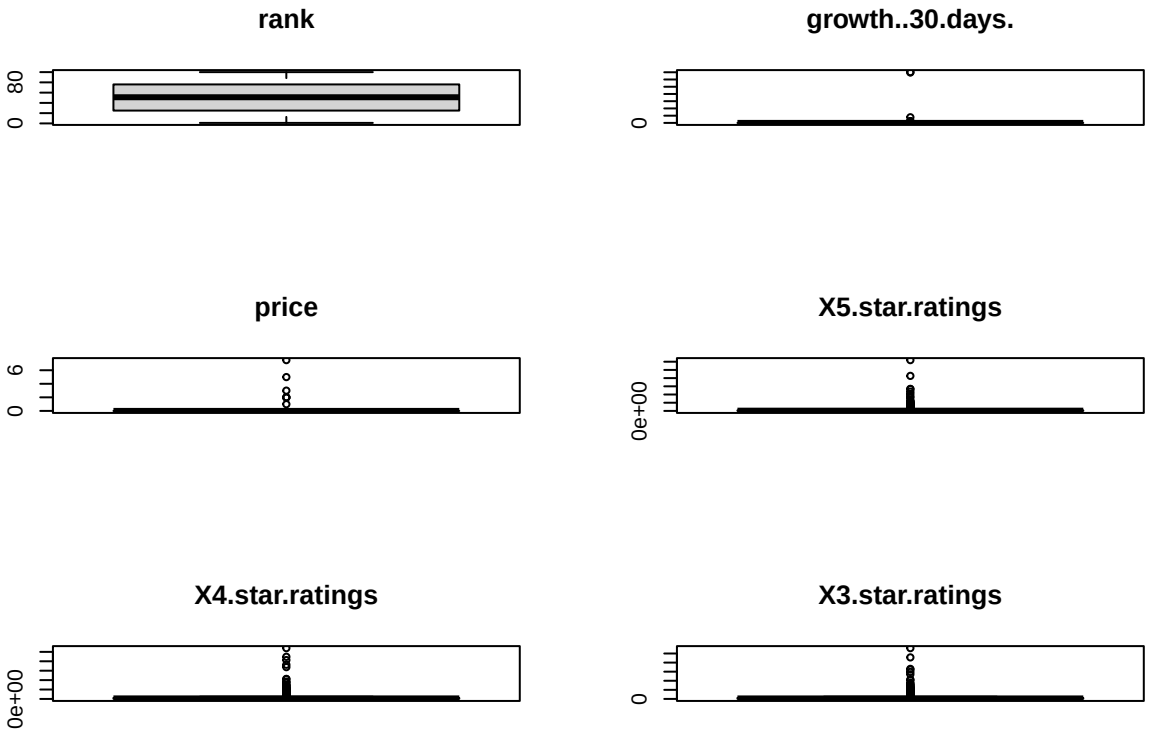
```
## [1] 1814999  893674  725429  383478  375159  217044
```

```
out_X1 <- boxplot.stats(games$X1.star.ratings)$out
head(out_X1)
```

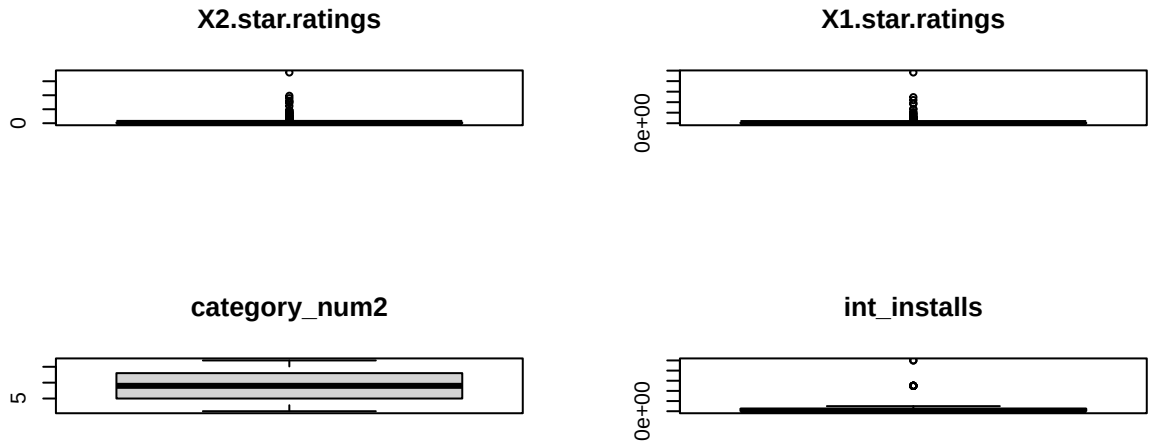
```
## [1] 9654037 4945478 4397640 2088781 1045945  889631
```

Vemos una representación gráfica con boxplot de las variables númericas para comprobar si existen valores extremos.

```
par(mfrow=c(3,2))
for(i in 1:ncol(games)) {
  if (is.numeric(games[,i])){
    boxplot(games[,i], main = colnames(games)[i], width = 100)
  }
}
```



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



Se considera valores extremos a aquellos valores cuando se encuentra alejado 3 desviaciones estándar con respecto a la media. Por ello, en muchos trabajos se utiliza la representación de los datos mediante gráficos de cajas (boxplots), con el objetivo de detectar dichos outliers.

Gráficamente vemos que las variables con valores muy por encima de la medía son paid, que no se puede considerar outliers ya que la mayoría de lo juegos son gratuitos, también int_install está muy por encima de la media pero no parece ser incorrecto porque es posible que hayas juegos con más megas que otros.

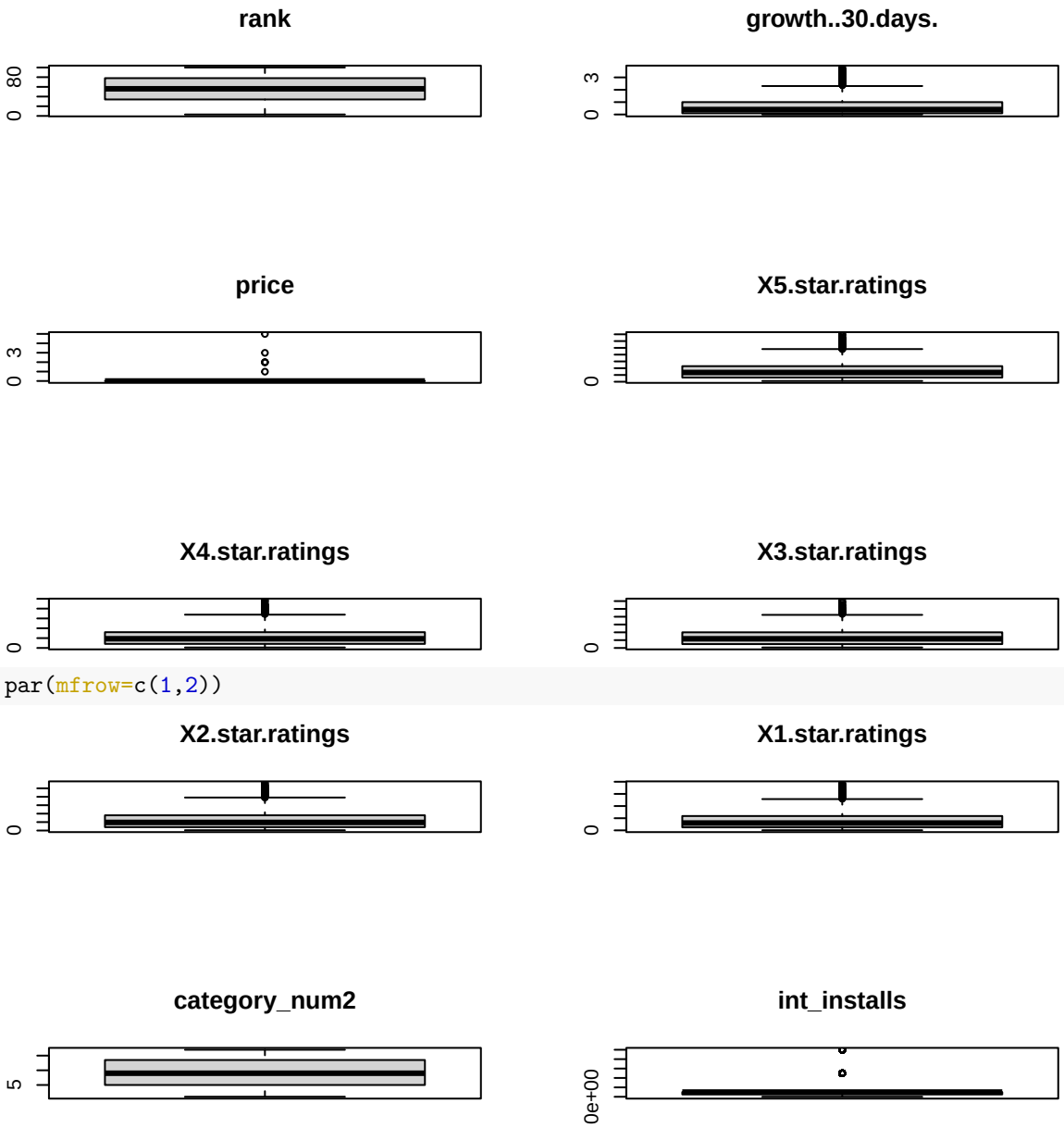
Las variables que si es posible que sean erroneos porque con fines mal intensionados se puede haber añadido excesivas valaraciones a un juego por medios automaticos (Rotos) de forma que se pueden haber valores extemos estas son: - growth..30.days. - X5.star.ratings
- X4.star.ratings
- X3.star.ratings - X2.star.ratings
- X1.star.ratings

Es por ello, que hemos decidido elinminar estos valores.

```
filas_old <-nrow(games)
games <- games[-which(games$growth..30.days. %in% out_growth),]
games <- games[-which(games$X5.star.ratings %in% out_X5),]
games <- games[-which(games$X4.star.ratings %in% out_X4),]
games <- games[-which(games$X3.star.ratings %in% out_X3),]
games <- games[-which(games$X2.star.ratings %in% out_X2),]
newgames <- games[-which(games$X1.star.ratings %in% out_X1),]
filas_new <-nrow(newgames)
```

Una vez eliminados los outlier, comprobamos gráficamente como se distribuyen los datos de estas variables.

```
par(mfrow=c(3,2))
for(i in 1:ncol(newgames)) {
  if (is.numeric(newgames[,i])){
    boxplot(newgames[,i], main = colnames(newgames)[i], width = 100)
  }
}
```




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

Hemos pasado de 1730 filas a 1296 filas.

Por último, exportamos el nuevo dataset a un nuevo csv.

```
my.newfile <- "newgame.csv"
write.csv(newgames, file=my.newfile, row.names = FALSE)
```

---

4. Análisis de los datos.

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

4.2. Comprobación de la normalidad y homogeneidad de la varianza.

4.3. 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.

---

5. Representación de los resultados a partir de tablas y gráficas.

---

6. Resolución del problema.

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