Universidad Nacional Autónoma de México

Facultad de Ciencias, 2024 - 1

Almacenes y Minería de Datos

Práctica 07. Análisis de Datos

PostgresandoesoSQLazos

Integrantes

Adrian Aguilera Moreno	421005200
Marco Antonio Rivera Silva	318183583
Sebastían Alejandro Gutiérrez Medina	318287021
Israel Hernández Dorantes	318206604
Alejandra Ortega García	420002495

1. Investiga los siguientes comandos de R o conceptos

- Media aritmética, geométrica y armónica

  - La **media aritmética** es el promedio de un conjunto de números. Se calcula sumando todos los números en el conjunto y luego dividiendo esta suma por la cantidad de números en el conjunto.

    Su formula es la siguiente:

    $$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

  - La **media geométrica** es una medida de tendencia central que se utiliza para calcular el promedio de un conjunto de números de una manera diferente a la media aritmética. Se calcula multiplicando todos los números en el conjunto y luego tomando la raíz enésima del producto, donde n es la cantidad de números en el conjunto.

    Su formula es la siguiente:

    $$M_G = \left(\prod_{i=1}^{n} x_i \right)^{\frac{1}{n}}$$

  - La **media armónica** es otra medida de tendencia central que se utiliza para calcular el promedio de un conjunto de números de una manera diferente a la media aritmética y la media geométrica. Se calcula dividiendo la cantidad de números en el conjunto entre la suma inversa de los valores recíprocos de esos números.

    Su formula es la siguiente:

    $$M_A = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

- stem(data), bagplot(data). ¿En que sentido difieren?

  En general, stem() es una buena opción para visualizar distribuciones de datos discretas de forma detallada. bagplot() es una buena opción para visualizar distribuciones de datos continuas de forma resistente a los valores atípicos.

  - stem() muestra la distribución de datos de forma más detallada que bagplot().
  - stem() muestra los valores individuales de los datos, mientras que bagplot() sólo muestra la media, la mediana y el rango intercuartil.
  - bagplot() es más resistente a los valores atípicos que stem(). bagplot() crea bolsas de datos aleatorios, por lo que los valores atípicos tienen menos influencia en la distribución de la bolsa.
  - stem() es más adecuada para distribuciones de datos discretas, mientras que bagplot() es más adecuada para distribuciones de datos continuas. stem() funciona bien para datos que pueden representarse como números enteros, mientras que bagplot() funciona bien para datos que pueden representarse como números reales.

2. ¿Cuál es el tipo del siguiente vector? ¿A que se debe esto? $q_2 \leftarrow (1, 2, 3, 4, cinco")$

En R es de tipo "character" (caracter). En R, cuando creas un vector que contiene elementos de diferentes tipos de datos, R intenta convertir todos los elementos a un tipo de datos común. En este caso, dado que uno de los elementos es una cadena de caracteres (string) "cinco", R convierte todos los elementos a caracteres para que todos los elementos del vector sean del mismo tipo.

3. Problema 1

Crear un vector de tamaño 10000 con valores entre -5000 y 5000 donde probara las funciones Min(), 1stQu(), Median(), Mean(), 3rd Qu(), Max() y las guardarás en este orden en un vector para comprobar mediante una función que es igual a la ejecución de la función summary() con el mismo vector. El archivo generado lo llamaran problema1.R.
Usando el siguiente código resolvimos el problema:

```
#problema1.R
vector <- sample(-5000:5000,10000,replace =TRUE)
print(vector)

min <- min(vector)
firstQ <- quantile(vector,0.25)
median <- median(vector)
mean <- mean(vector)
thirdQ <- quantile(vector,0.75)
max <- max(vector)
resultados <- c(min,firstQ,median,mean,thirdQ,max)

print(resultados)
summary(vector)
```

Como podemos ver, la salida de de comparar ambos vectores es TRUE por lo que podemos concluir que son funciones equivalentes, sin embargo summary() es mucho mas fácil de usar.

```
> print(resultados)
                  25%                                  75%
-4998.0000 -2479.0000    38.0000    12.4866  2469.0000  4999.0000
> summary(vector)
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
-4998.00 -2479.00    38.00    12.49  2469.00  4999.00
> print(resultados == summary(vector))
      25%            75%
TRUE TRUE TRUE TRUE TRUE TRUE
> S
```

Figure 1: Comparación de salidas

4. Problema 2

De los datasets grupoA.csv y grupoB.csv, deberán realizar lo siguiente:

- Muestra la distribución de tipos sanguíneos de la variable Gr Sang mediante una gráfica de pay.

  - **Grupo A** Para generar la imagen usamos el código:

```
# Grafica de pay de GrupoA
conteosA <- table(grupoA$Gr_sang)
pie(conteosA, labels = paste(names(conteosA), "\n(", conteosA, ")",sep = ""),
    main = "Grupo A")
```

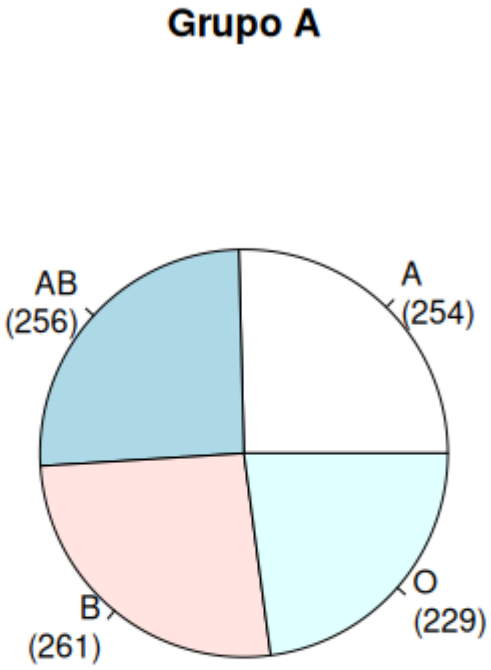


Figure 2: Gráfica de Pay de la distribución de grupos sanguíneos de A

  - **Grupo B** Para generar la imagen usamos el código:

```
# Grafica de pay de GrupoB
conteosB <- table(grupoB$Gr_sang)
pie(conteosB, labels = paste(names(conteosB), "\n(", conteosB, ")",sep = ""),
    main = "Grupo B")
```

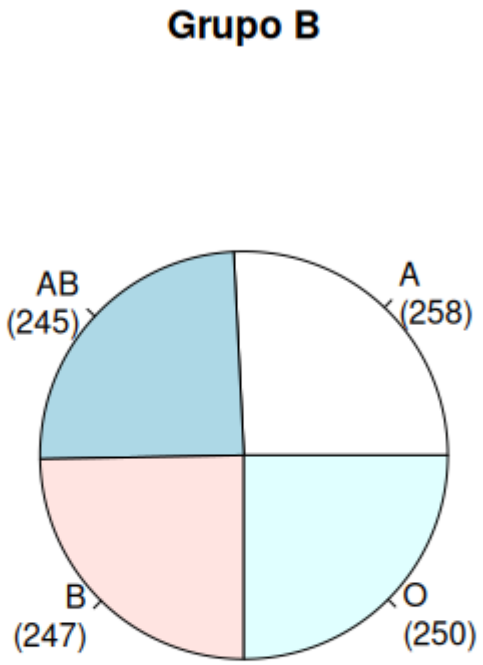


Figure 3: Gráfica de Pay de la distribución de grupos sanguíneos de B

- Muestra los datos de la variable Estatura en un histograma para cada conjunto.
  - **Grupo A** Para generar la imagen usamos el código:

```
# Histograma de variable estaturas del grupo A
hist(grupoA$Estatura, main = "Histograma estaturas Grupo A")
```

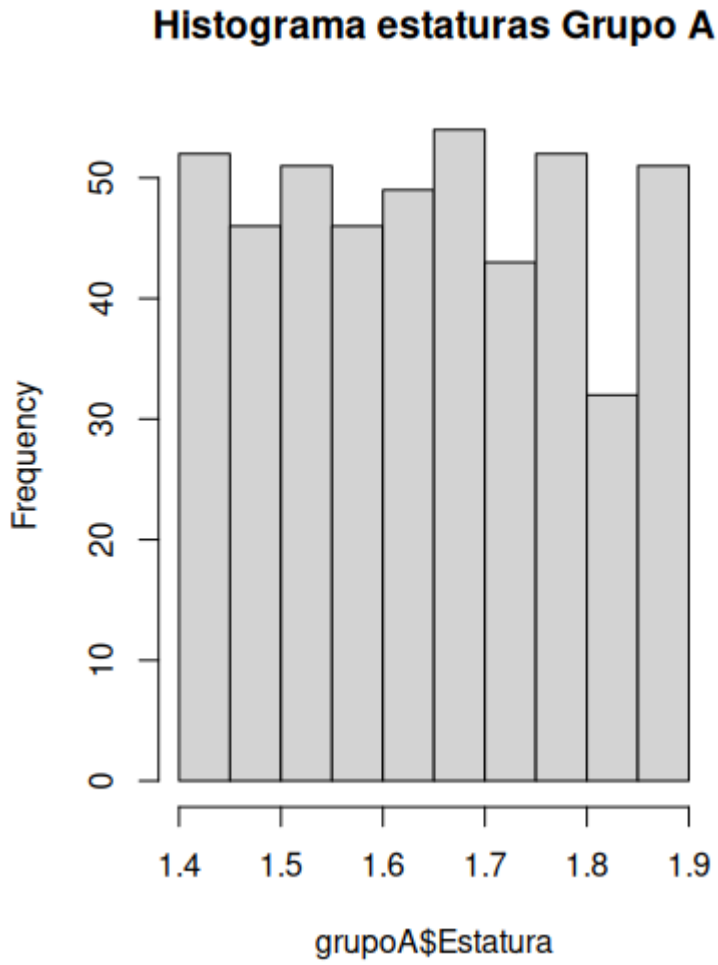


Figure 4: Histograma de la estatura de A

- **Grupo B** Para generar la imagen usamos el código:

```
# Histograma de variable estaturas del grupo B
hist(grupoB$Estatura, main = "Histograma estaturas Grupo B")
```

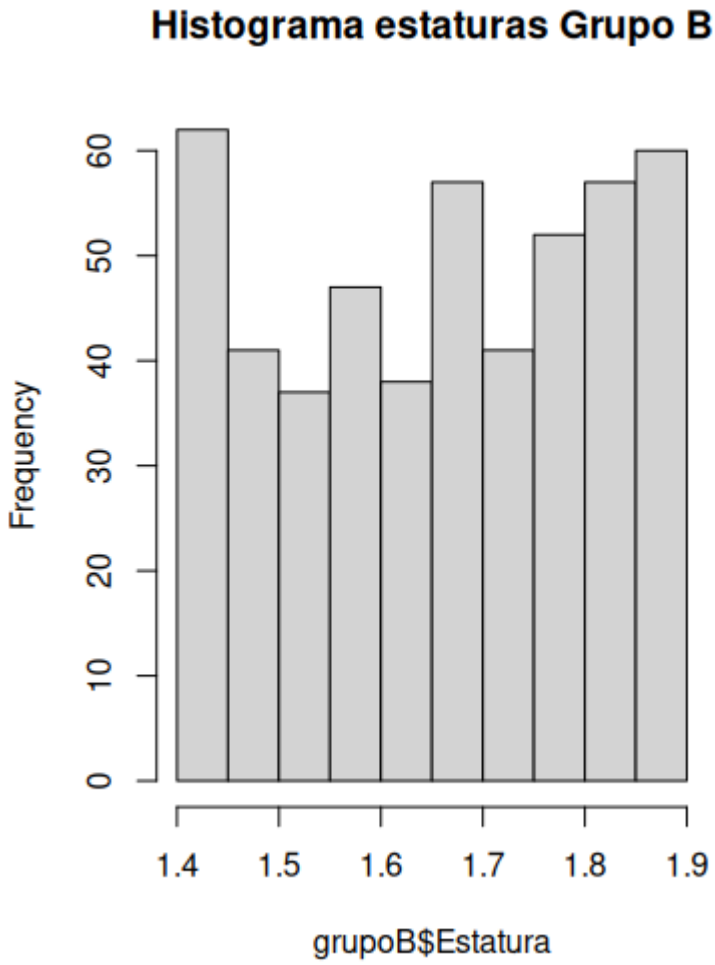


Figure 5: Histograma de la estatura de A

- Busca si hay datos atípicos en la variable edad. ¿Hay valores atípicos en el grupo A y grupo B? (Hint: Pueden utilizar una gráfica de caja).

  Para responder la pregunta usamos el código:

```r
# Crear un gráfico de caja del grupo A
boxplot(grupoA$Edad, main = "Gráfico de Caja Grupo A")

# Agregar identificadores a los valores atípicos
outliers <- boxplot(grupoA$Edad, plot = FALSE)$out
if (length(outliers) > 0) {
  points(rep(1, length(outliers)), outliers, col = "red", pch = 19)
  cat("Valores atípicos:", outliers, "\n")
} else {
  cat("No hay valores atípicos en el conjunto de datos.\n")
}
```

```r
# Crear un gráfico de caja del grupo B
boxplot(grupoB$Edad, main = "Gráfico de Caja Grupo B")

# Agregar identificadores a los valores atípicos
outliers <- boxplot(grupoB$Edad, plot = FALSE)$out
if (length(outliers) > 0) {
  points(rep(1, length(outliers)), outliers, col = "red", pch = 19)
  cat("Valores atípicos:", outliers, "\n")
} else {
  cat("No hay valores atípicos en el conjunto de datos.\n")
}
```

El resultado fue el siguiente:

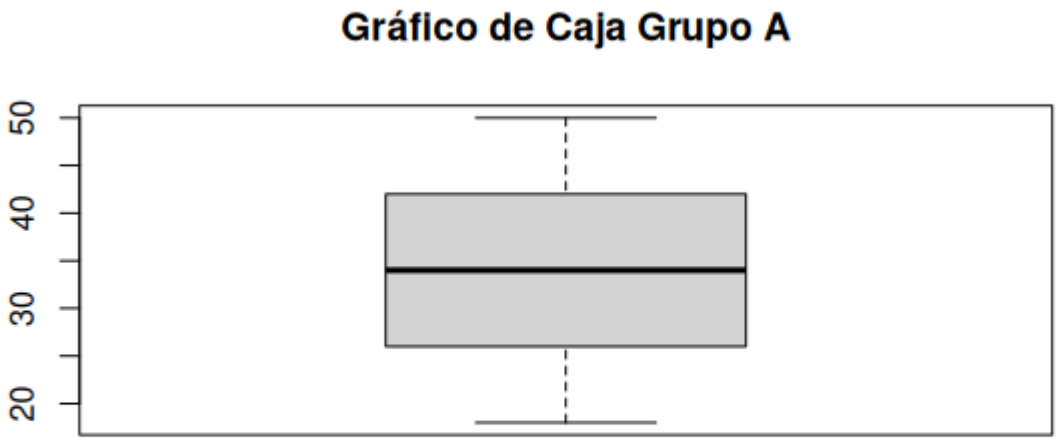


Figure 6: Grafica de caja de la edad de A

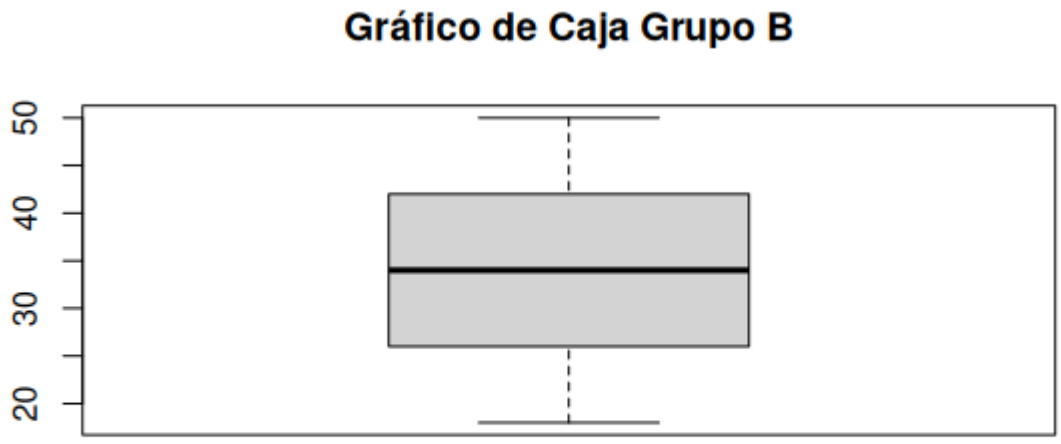


Figure 7: Grafica de caja de la edad de B

También la salida fue que ninguno de los dos grupos tiene valores atípicos en la variable edad.

- ¿Cual grupo tiene un promedio de altura mayor? Y ¿Cual tiene una altura mediana menor?

  Usamos el siguiente código para responder a la pregunta:

```
# Promedio de estaturas de A
meanEstaturasA <- mean(grupoA$Estatura, na.rm = TRUE)
# Promedio de estaturas de B
meanEstaturasB <- mean(grupoB$Estatura, na.rm = TRUE)

print(meanEstaturasA)
print(meanEstaturasB)
```

  El resultado fue el siguiente:

```
> print(meanEstaturasA)
[1] 1.648655
> print(meanEstaturasB)
[1] 1.66372
>
```

Figure 8: Histograma de la estatura de A

Por lo que podemos ver que el grupo B tienen un promedio de altura mayor. Por otro lado, para contestar la segunda pregunta, utilizamos el código:

```r
# Mediana de estaturas de A
medianEstaturasA <- median(grupoA$Estatura, na.rm = TRUE)
# Mediana de estaturas de B
medianEstaturasB <- median(grupoB$Estatura, na.rm = TRUE)

print(meanEstaturasA)
print(meanEstaturasB)
```

El resultado fué el siguiente:

```
> # Mediana de estaturas de A
> medianEstaturasA <- median(grupoA$Estatura, na.rm = TRUE)
> # Mediana de estaturas de B
> medianEstaturasB <- median(grupoB$Estatura, na.rm = TRUE)
> print(meanEstaturasA)
[1] 1.648655
> print(meanEstaturasB)
[1] 1.66372
>
```

Figure 9: Histograma de la estatura de A

Por lo que podemos concluir que el grupo A tiene una altura mediana menor.

5. Análisis de datos sobre el dataset vgsales.csv

Deberán hacer el análisis de datos sobre el dataset vgsales.csv. Donde analizaran cada columna. Ver si existe alguna correlación entre los atributos. Si hay datos atípicos. Ver que valores se repiten más. Si hay valores faltantes, o errores en los datos. Deberán apoyar este análisis con las gráficas. Todo el análisis que hicieron deberán de guardarlo en un archivo llamado análisis.R. Es importante que hagan un análisis complejo, ya que nos servirá para la siguiente práctica.

Después de cargar nuestro conjunto de datos procedemos a realizar un summary sobre el dataset:

```
> summary(vgsales)
      Rank           Name              Platform              Year
 Min.   :    1   Length:16598       Length:16598       Min.   :1980
 1st Qu.: 4151   Class :character   Class :character   1st Qu.:2003
 Median : 8300   Mode  :character   Mode  :character   Median :2007
 Mean   : 8301                                         Mean   :2006
 3rd Qu.:12450                                         3rd Qu.:2010
 Max.   :16600                                         Max.   :2020
                                                       NA's   :289
    Genre            Publisher            NA_Sales          EU_Sales
 Length:16598       Length:16598       Min.   : 0.0000   Min.   : 0.0000
 Class :character   Class :character   1st Qu.: 0.0000   1st Qu.: 0.0000
 Mode  :character   Mode  :character   Median : 0.0800   Median : 0.0200
                                       Mean   : 0.2647   Mean   : 0.1467
                                       3rd Qu.: 0.2400   3rd Qu.: 0.1100
                                       Max.   :41.4900   Max.   :29.0200
                                       NA's   :5         NA's   :5
    JP_Sales          Other_Sales        Global_Sales
 Min.   : 0.00000   Min.   : 0.00000   Min.   :-23.1000
 1st Qu.: 0.00000   1st Qu.: 0.00000   1st Qu.:  0.0600
 Median : 0.00000   Median : 0.01000   Median :  0.1700
 Mean   : 0.07778   Mean   : 0.04807   Mean   :  0.5348
 3rd Qu.: 0.04000   3rd Qu.: 0.04000   3rd Qu.:  0.4700
 Max.   :10.22000   Max.   :10.57000   Max.   : 82.7400
 NA's   :5          NA's   :5          NA's   :5
>
```

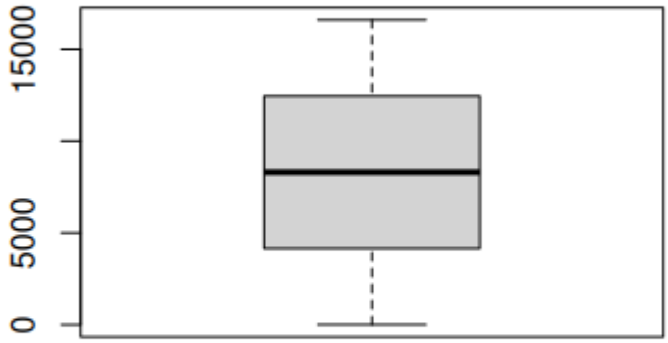
Mostramos los primeros datos del dataset con el siguiente código:

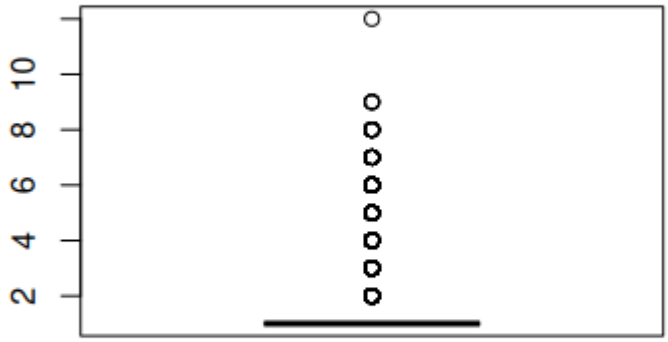
```
# Vemos las primeras lineas del dataset
head(vgsales)
```

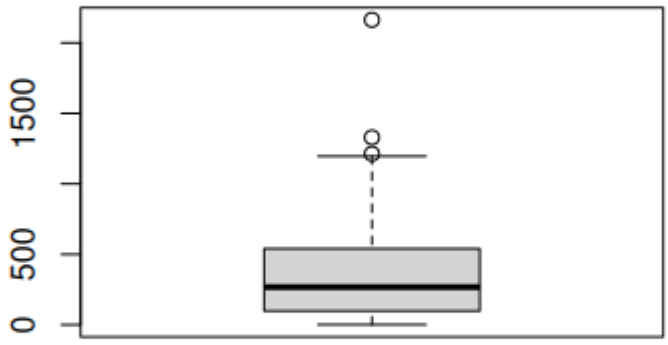
```
> # Vemos las primeras lineas del dataset
> head(vgsales)
  Rank                     Name Platform Year         Genre Publisher NA_Sales EU_Sales JP_Sales
1    1               Wii Sports       WI 2006        Sports   intendo    41.49    29.02     3.77
2    2        Super Mario Bros.      NES 1985      Platform  Nintendo    29.08     3.58     6.81
3    3           Mario Kart Wii      wII 2008        Racing  Nintendo    15.85    12.88     3.79
4    4        Wii Sports Resort       wi 2009        Sports   intendo    15.75    11.01     3.28
5    5 Pokemon Red/Pokemon Blue       GB 1996 Role-Playing   intendo    11.27     8.89    10.22
6    6                   Tetris       GB 1989        Puzzle  Nintendo    23.20     2.26     4.22
  Other_Sales Global_Sales
1        8.46        82.74
2        0.77        40.24
3        3.31        35.82
4        2.96        33.00
5        1.00        31.37
6        0.58        30.26
>
```

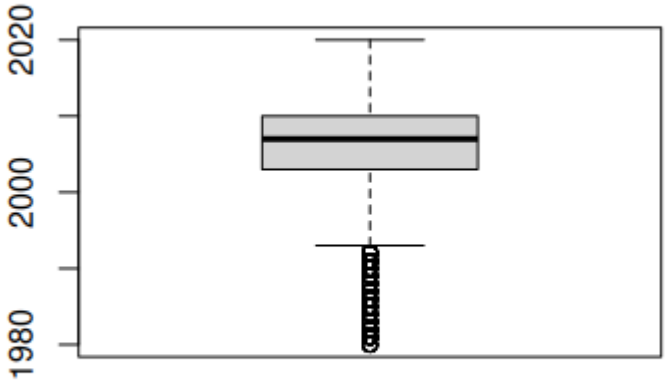
Hacemos gráficas de caja para cada una de las columnas para checar si hay datos anómalos.
Usamos el siguiente código

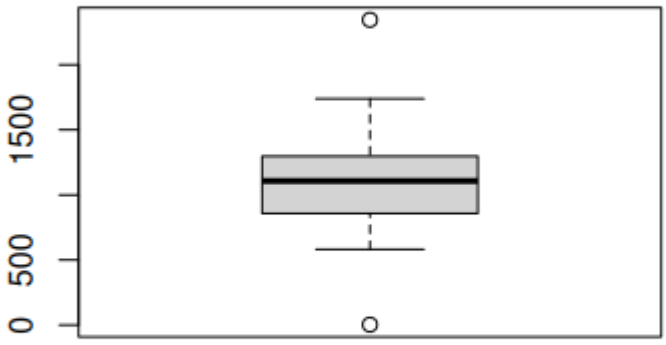
```r
# Boxplot para verificar valoresa anomalos en cada columna (Solo para variables numericas)
# Para variables categoricas añadimos tambien la función table
boxplot(vgsales$Rank)
boxplot(table(vgsales$Name))
boxplot(table(vgsales$Platform))
boxplot(vgsales$Year)
boxplot(table(vgsales$Genre))
boxplot(table(vgsales$Publisher))
boxplot(vgsales$NA_Sales)
boxplot(vgsales$EU_Sales)
boxplot(vgsales$JP_Sales)
boxplot(vgsales$Other_Sales)
boxplot(vgsales$Global_Sales)
```

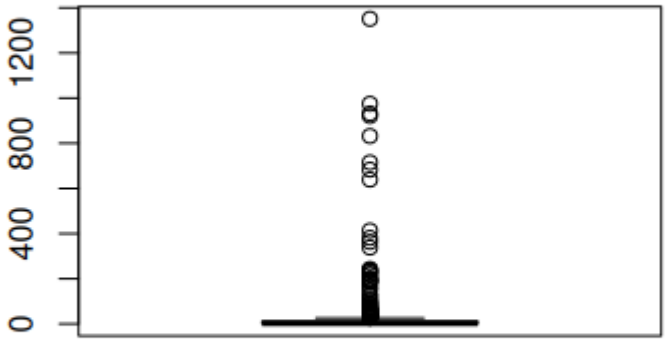


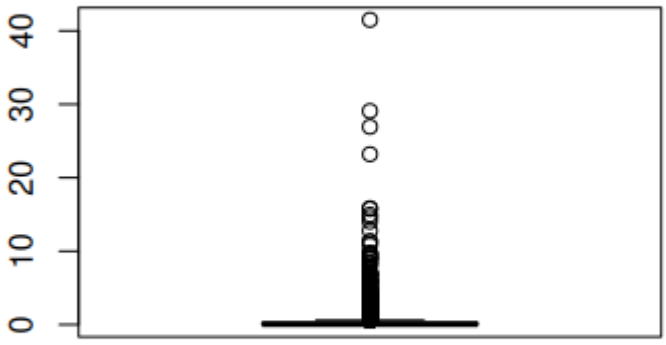

2
4
6
8
10

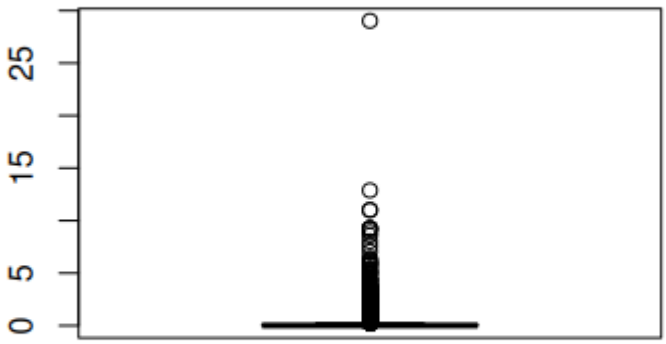

0
500
1500

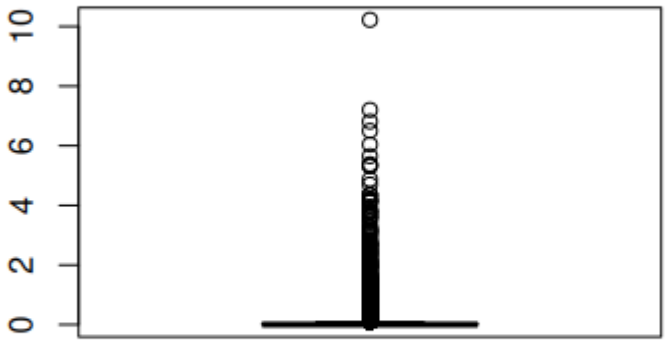

2020
2000
1980

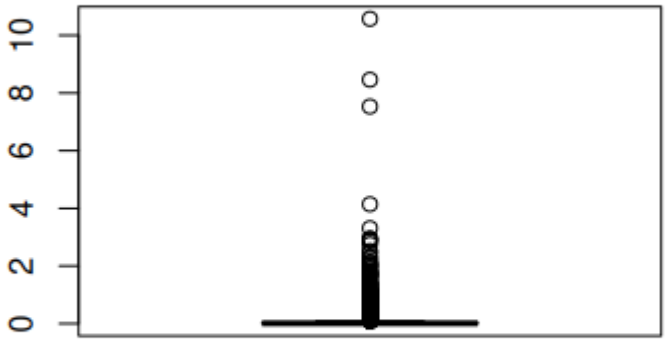

0
500
1500

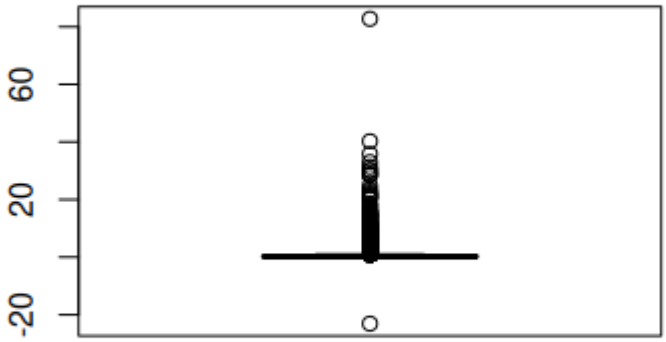

0
400
800
1200


0
10
20
30
40


0
5
15
25


0
2
4
6
8
10

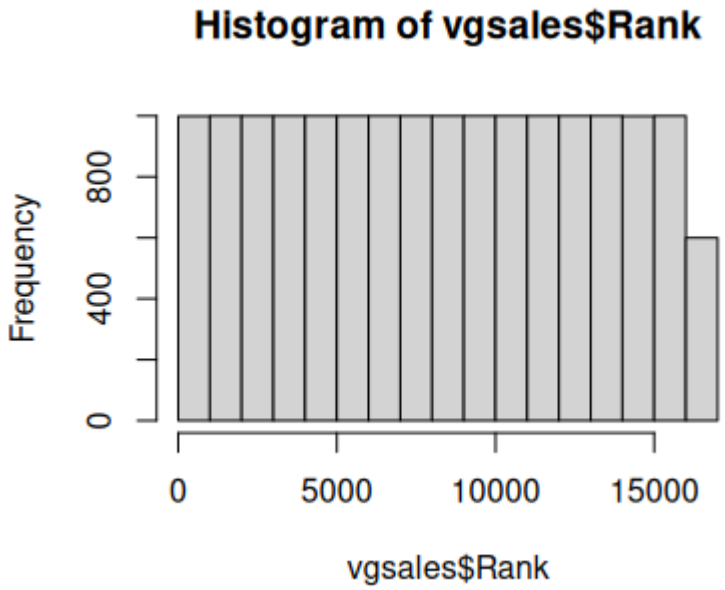


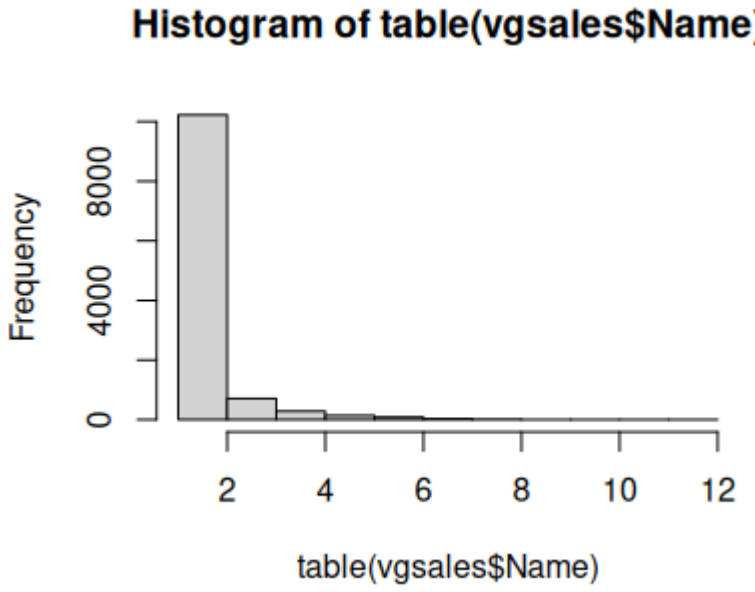


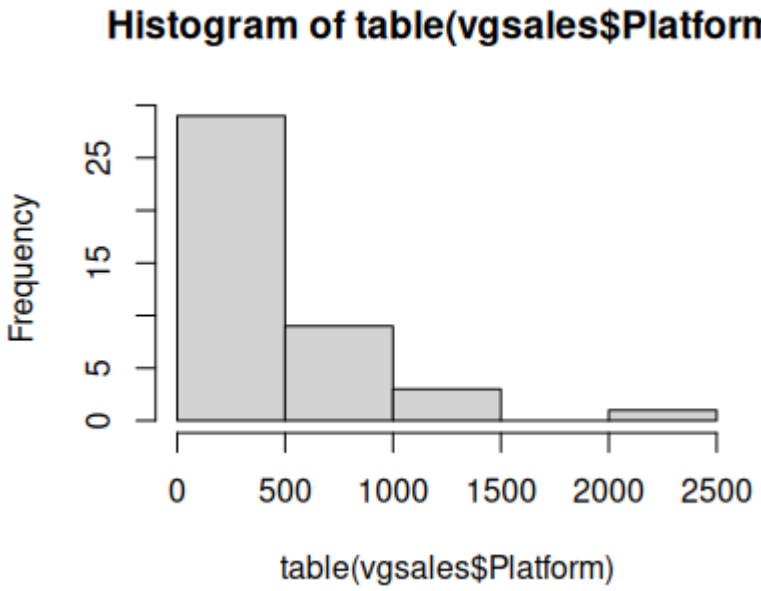
Después obtuvimos histogramas para cada columna con el siguiente código:

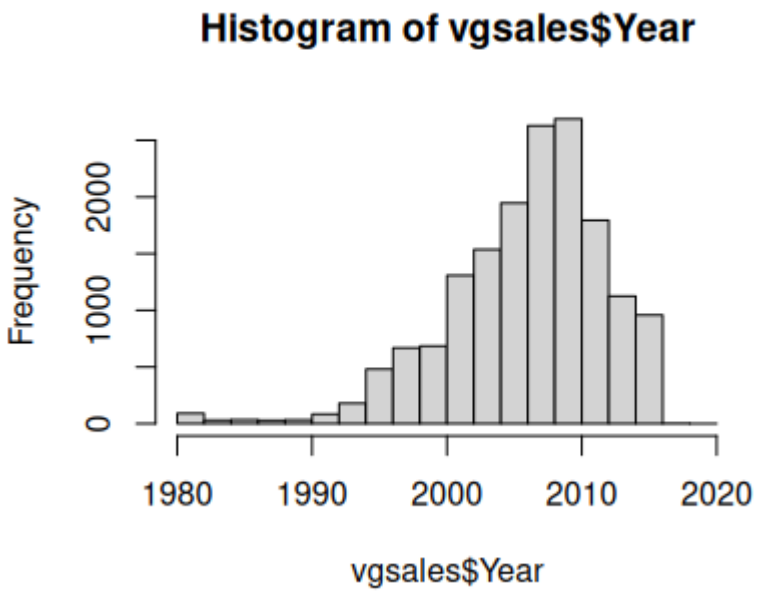
```
#Hacemos un histograma por cada columna:
#Para variables categoricas añadimos tambien la función table
hist(vgsales$Rank)
hist(table(vgsales$Name))
hist(table(vgsales$Platform))
hist(vgsales$Year)
hist(table(vgsales$Genre))
```

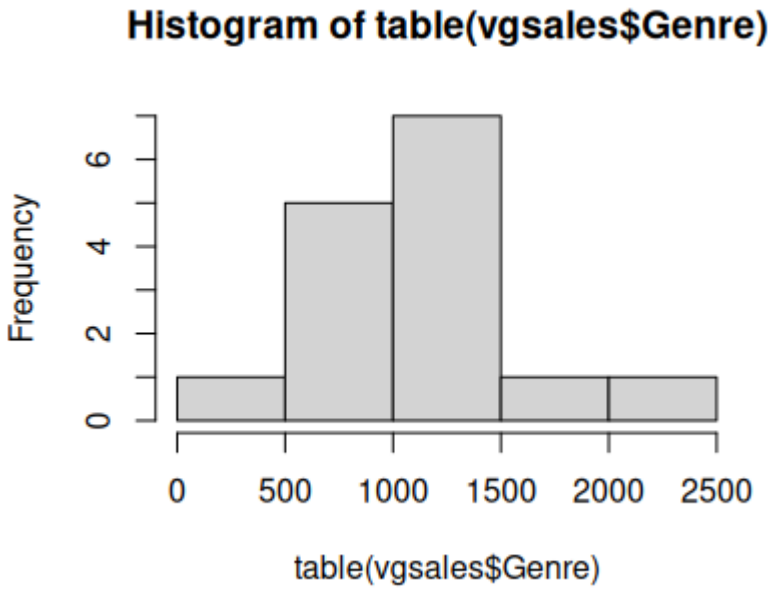
```
hist(table(vgsales$Publisher))
hist(vgsales$NA_Sales)
hist(vgsales$EU_Sales)
hist(vgsales$JP_Sales)
hist(vgsales$Other_Sales)
hist(vgsales$Global_Sales)
```

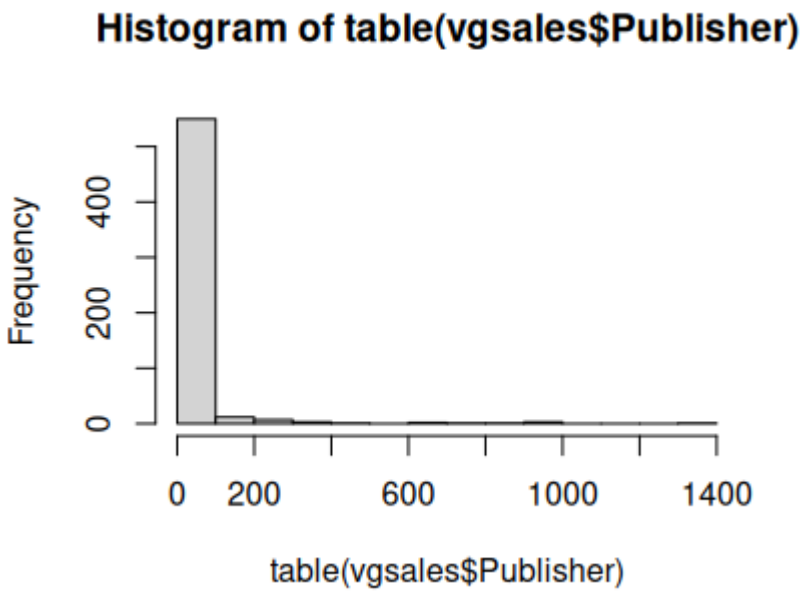


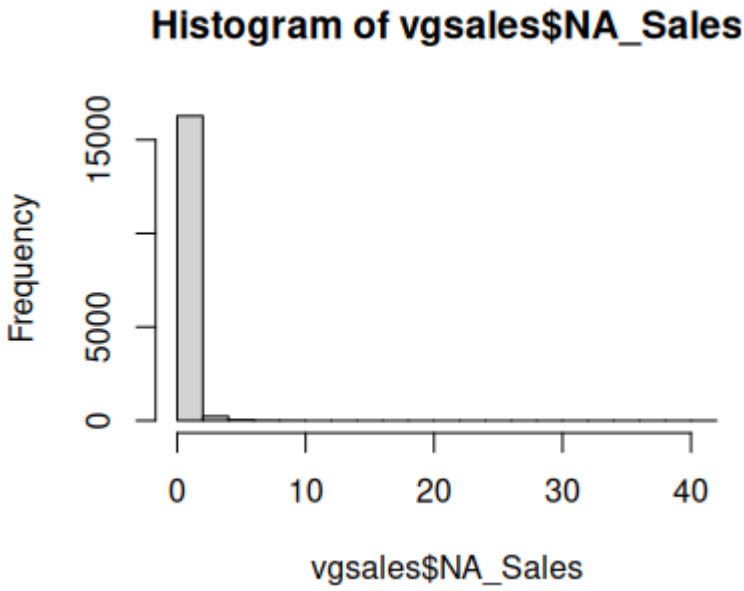


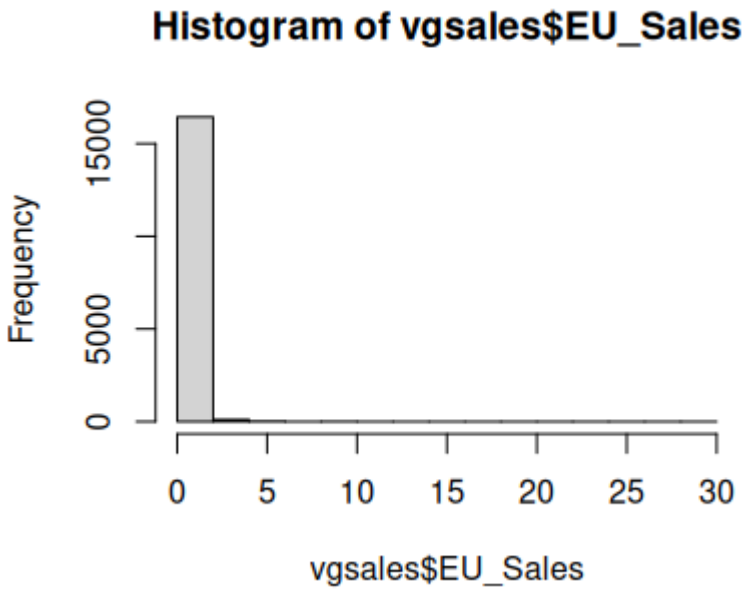

Histogram of table(vgsales$Platform)
Frequency
0
5
15
25
0
500
1000
1500
2000
2500
table(vgsales$Platform)

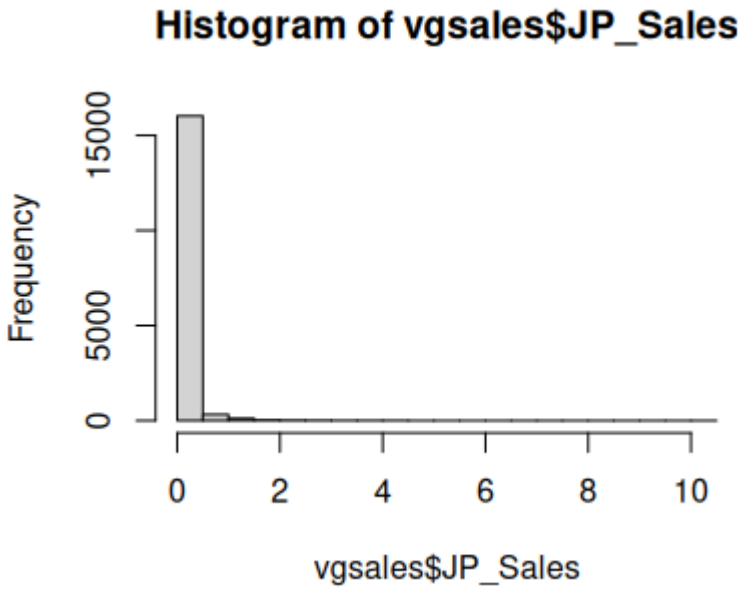

Histogram of vgsales$Year
Frequency
0
1000
2000
1980
1990
2000
2010
2020
vgsales$Year

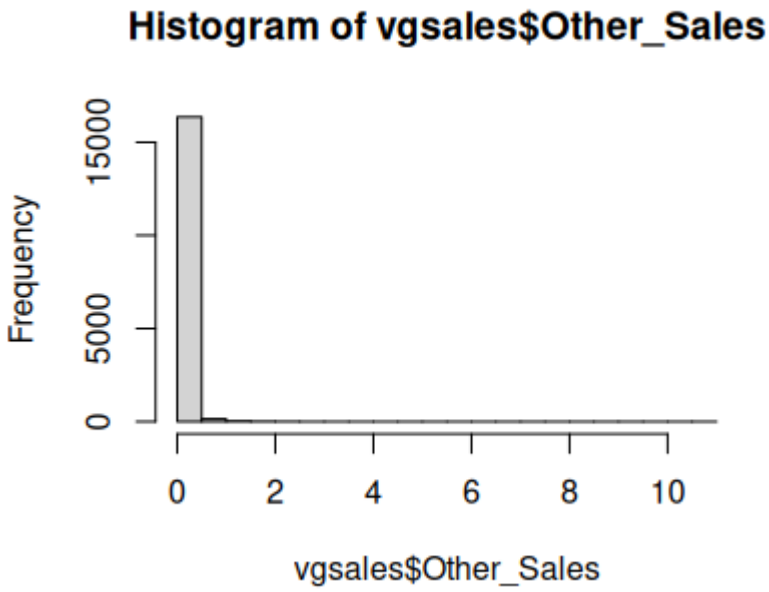

Histogram of table(vgsales$Genre)
Frequency
0
2
4
6
0
500
1000
1500
2000
2500
table(vgsales$Genre)

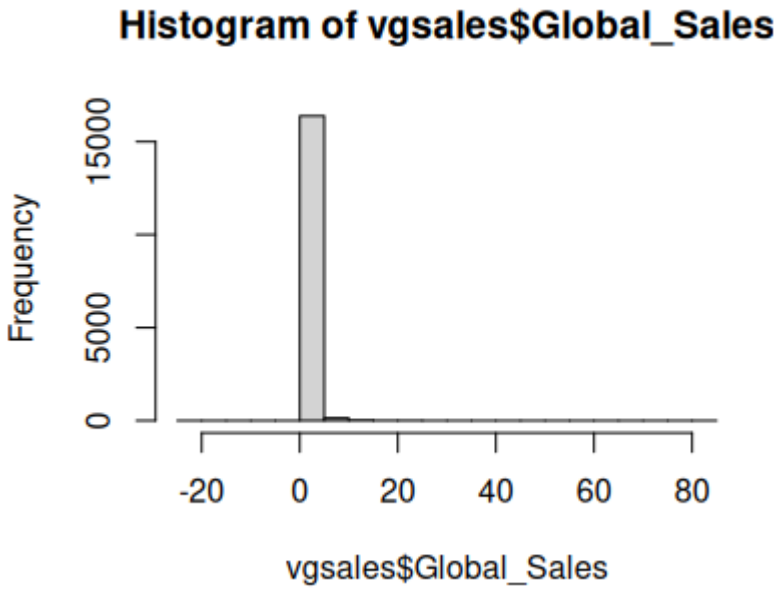

Histogram of table(vgsales$Publisher)
Frequency
0
200
400
0
200
600
1000
1400
table(vgsales$Publisher)


Histogram of vgsales$NA_Sales
Frequency
0
5000
15000
0
10
20
30
40
vgsales$NA_Sales


Histogram of vgsales$EU_Sales
Frequency
0
5000
15000
0
5
10
15
20
25
30
vgsales$EU_Sales

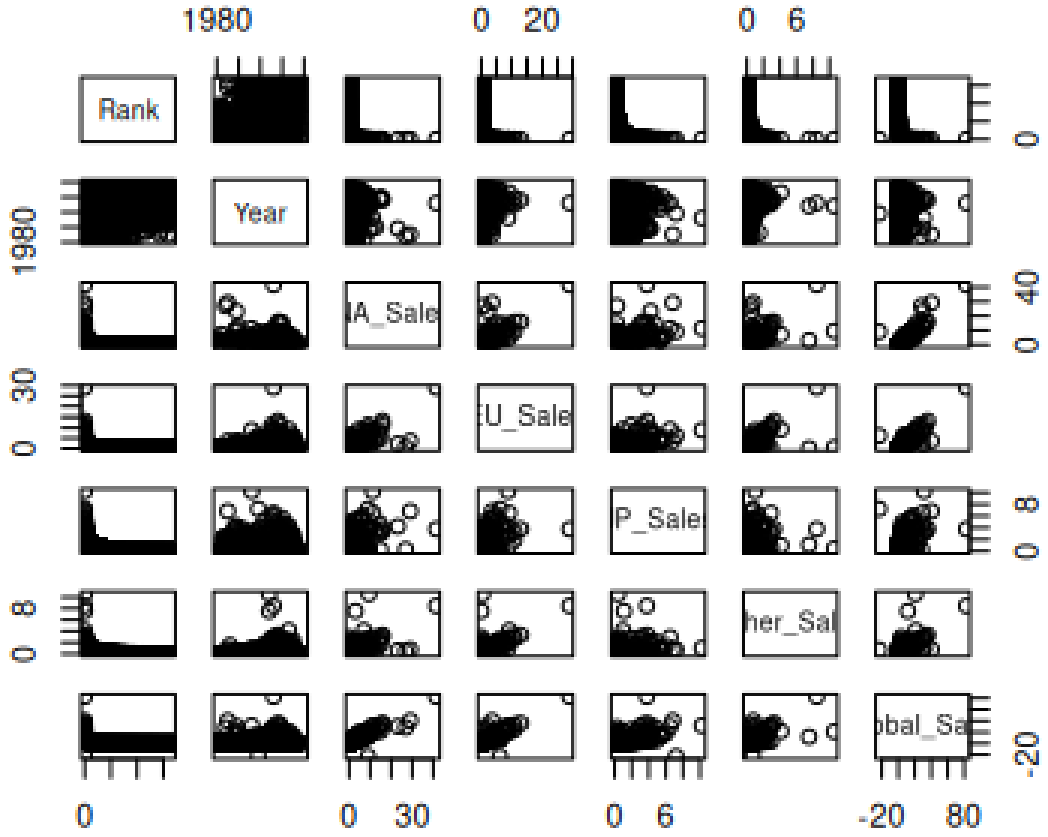

Histogram of vgsales$JP_Sales
Frequency
0
5000
15000
0
2
4
6
8
10
vgsales$JP_Sales


Histogram of vgsales$Other_Sales
Frequency
0
5000
15000
0
2
4
6
8
10
vgsales$Other_Sales



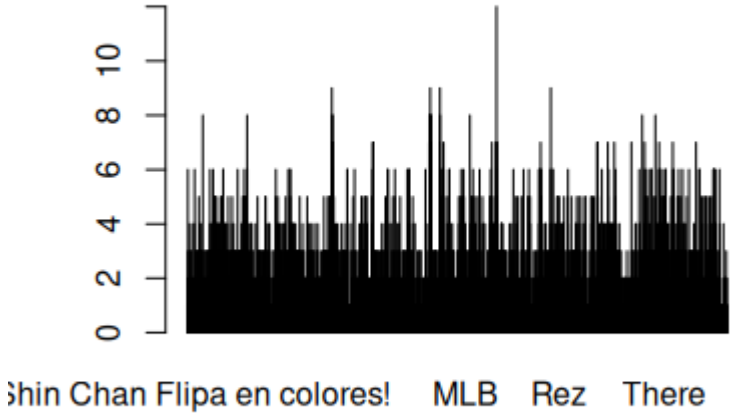
Checamos la dispersión entre las variables númericas con el siguiente código:

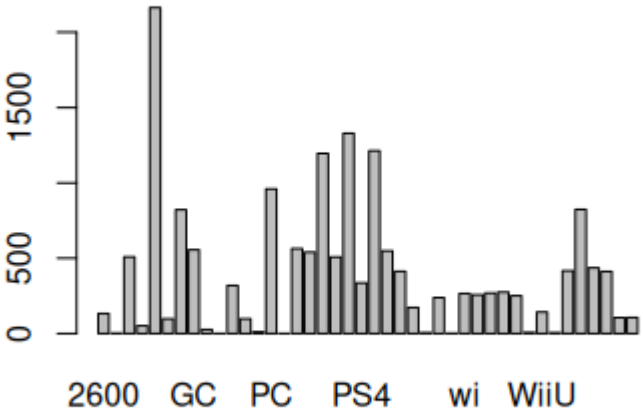
```
# Gráfico de dispersión para atributos numéricos
pairs(vgsales[, sapply(vgsales, is.numeric)])
```

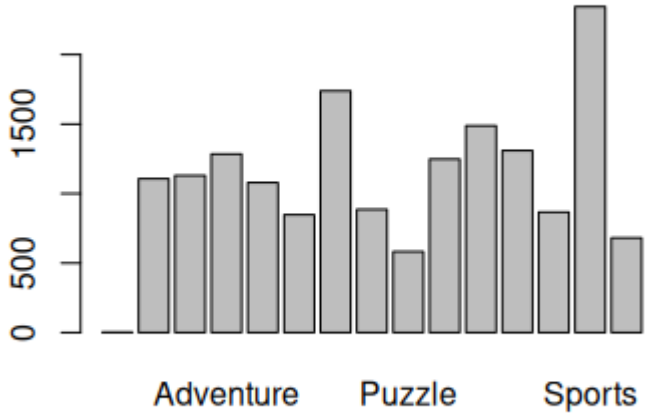


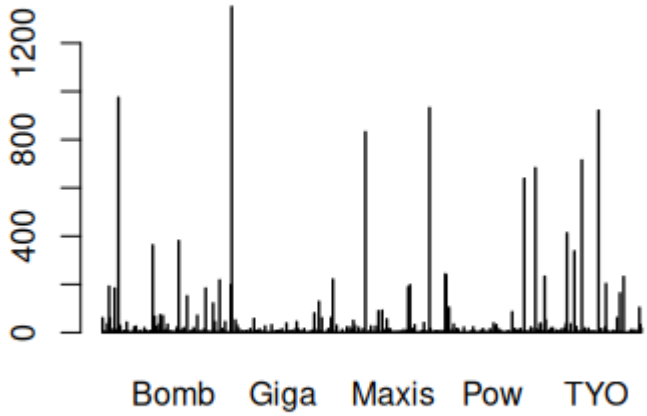
Generamos gráfica de barras para las variables categóricas con el siguiente código:

```
# Valores más comunes en una columna específica (variables categoricas)
barplot(table(vgsales$Name))
barplot(table(vgsales$Platform))
barplot(table(vgsales$Genre))
barplot(table(vgsales$Publisher))
```


0 2 4 6 8 10
Shin Chan
Flipa en colores!
MLB
Rez
There


0 500 1500
2600
GC
PC
PS4
wi
WiiU





Después checamos si había datos faltantes en cada columna con el siguiente comando:

```
# Detección de valores faltantes
cat("Valores faltantes en el conjunto de datos:", sum(is.na(data)), "\n")
print(sum(is.na(vgsales$Rank)))
print(sum(is.na(vgsales$Name)))
print(sum(is.na(vgsales$Platform)))
print(sum(is.na(vgsales$Year)))
print(sum(is.na(vgsales$Genre)))
```

```
print(sum(is.na(vgsales$Publisher)))
print(sum(is.na(vgsales$NA_Sales)))
print(sum(is.na(vgsales$EU_Sales)))
print(sum(is.na(vgsales$JP_Sales)))
print(sum(is.na(vgsales$Other_Sales)))
print(sum(is.na(vgsales$Global_Sales)))
```

```
> # Detección de valores faltantes
> cat("Valores faltantes en el conjunto de datos:", sum(is.na(data)), "\n")
Valores faltantes en el conjunto de datos: 314
> print(sum(is.na(vgsales$Rank)))
[1] 0
> print(sum(is.na(vgsales$Name)))
[1] 0
> print(sum(is.na(vgsales$Platform)))
[1] 0
> print(sum(is.na(vgsales$Year)))
[1] 289
> print(sum(is.na(vgsales$Genre)))
[1] 0
> print(sum(is.na(vgsales$Publisher)))
[1] 0
> print(sum(is.na(vgsales$NA_Sales)))
[1] 5
> print(sum(is.na(vgsales$EU_Sales)))
[1] 5
> print(sum(is.na(vgsales$JP_Sales)))
[1] 5
> print(sum(is.na(vgsales$Other_Sales)))
[1] 5
> print(sum(is.na(vgsales$Global_Sales)))
[1] 5
>
```

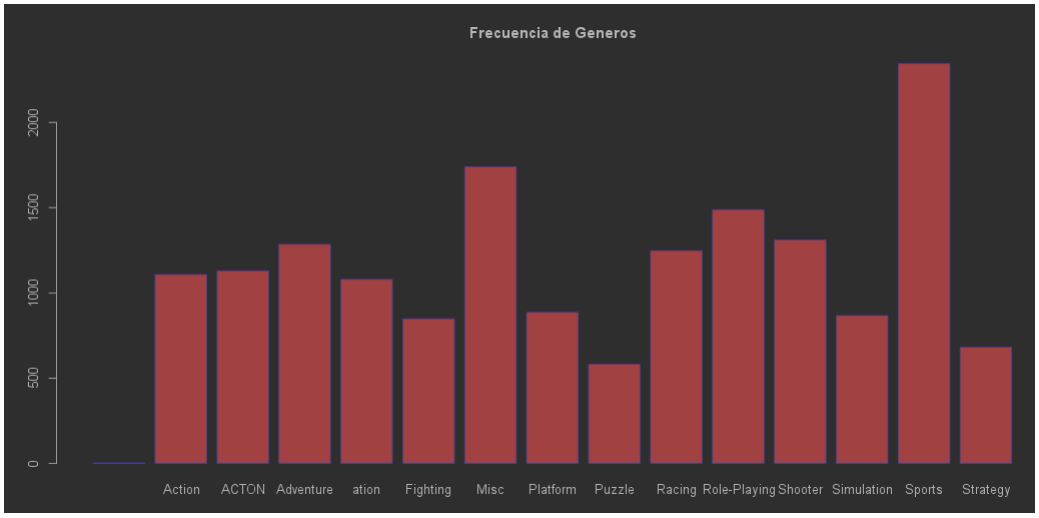
Además checamos si los valores de cantidades tenian errores (como cantidades negativas) con el siguiente código:

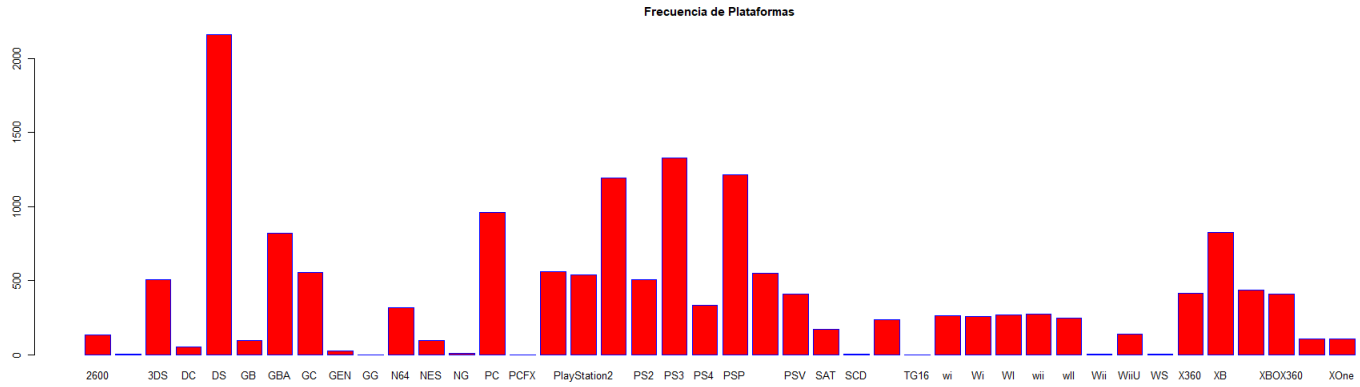
```
# Revisión de posibles errores en los datos (por ejemplo, valores negativos en ventas)
cat("Valores negativos en las ventas globales:", sum(vgsales$NA_Sales < 0), "\n")
cat("Valores negativos en las ventas globales:", sum(vgsales$EU_Sales < 0), "\n")
cat("Valores negativos en las ventas globales:", sum(vgsales$JP_Sales < 0), "\n")
cat("Valores negativos en las ventas globales:", sum(vgsales$Other_Sales < 0), "\n")
cat("Valores negativos en las ventas globales:", sum(vgsales$Global_Sales < 0), "\n")
```

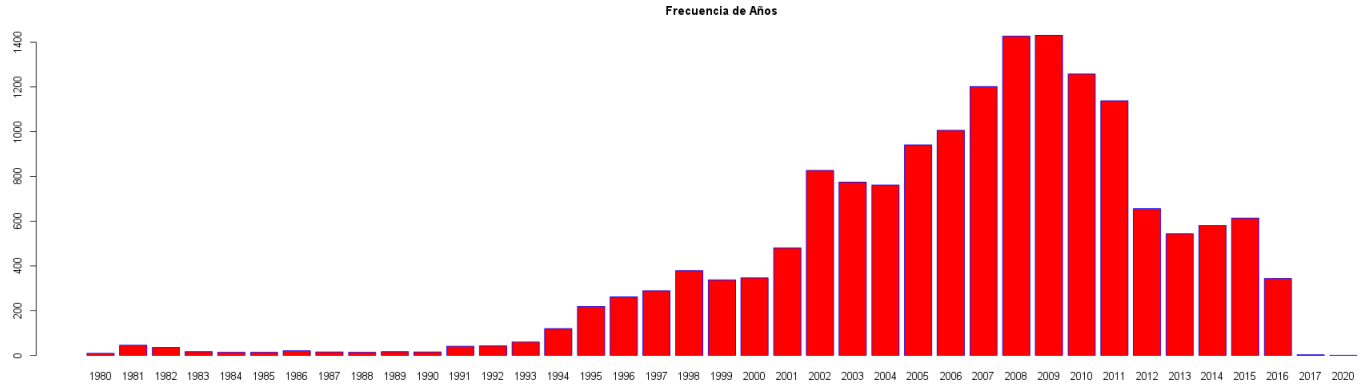
```
> # Revisión de posibles errores en los datos (por ejemplo, valores negativos en ventas)
> cat("Valores negativos en las ventas globales:", sum(vgsales$NA_Sales < 0), "\n")
Valores negativos en las ventas globales: NA
> cat("Valores negativos en las ventas globales:", sum(vgsales$EU_Sales < 0), "\n")
Valores negativos en las ventas globales: NA
> cat("Valores negativos en las ventas globales:", sum(vgsales$JP_Sales < 0), "\n")
Valores negativos en las ventas globales: NA
> cat("Valores negativos en las ventas globales:", sum(vgsales$Other_Sales < 0), "\n")
Valores negativos en las ventas globales: NA
> cat("Valores negativos en las ventas globales:", sum(vgsales$Global_Sales < 0), "\n")
Valores negativos en las ventas globales: NA
>
```

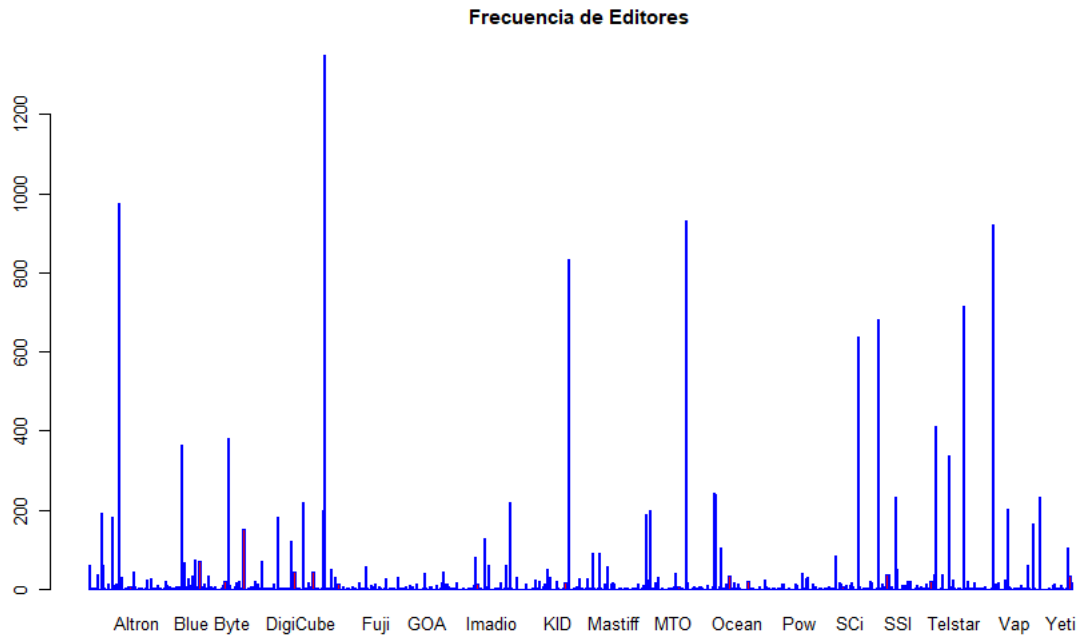
Para las frecuencias de utilizamos lo siguiente :

```r
# Calculo de frecuencias
barplot(table(videojuegos$Platform), main = "Frecuencia de Plataformas",
col = "red", border = "blue")
barplot(table(videojuegos$Year), main = "Frecuencia de Años",
col = "red", border = "blue")
barplot(table(videojuegos$Genre), main = "Frecuencia de Generos",
col = "red", border = "blue")
barplot(table(videojuegos$Publisher), main = "Frecuencia de Editores",
col = "red", border = "blue")
```




Frecuencia de Plataformas
0
500
1000
1500
2000
2600
3DS
DC
DS
GB
GBA
GC
GEN
GG
N64
NES
NG
PC
PCFX
PlayStation2
PS2
PS3
PS4
PSP
PSV
SAT
SCD
TG16
wi
Wi
WI
wii
wII
Wii
WiiU
WS
X360
XB
XBOX360
XOne


Frecuencia de Años
0
200
400
600
800
1000
1200
1400
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2020


Frecuencia de Editores
0
200
400
600
800
1000
1200
Altron
Blue Byte
DigiCube
Fuji
GOA
Imadio
KID
Mastiff
MTO
Ocean
Pow
SCi
SSI
Telstar
Vap
Yeti