Reporte de investigación

Oscar Elí Bonilla Morales

2022-05-30

Introducción

Sin duda la industria de los videojuegos ha crecido exponencialmente en los últimos años, la gran variedad de géneros y estilos han sido algunas de las principales causas por las cuales este tipo de entretenimiento ha cautivado a millones de jugadores alrededor del mundo. Al contar con un gran número de seguidores es posible obtener grandes cantidades de información la cual puede ser útil tanto para las empresas desarrolladoras como para los jugadores, quienes a través de esta información pueden considerar distintas estrategias, verificar errores dentro de los juegos, crear estadísticas sobre sus juegos favoritos o mejorar su rendimiento dentro del juego.

Un ejemplo de lo antes mencionado es el cómo los jugadores utilizan las estadísticas de ciertos personajes, estableciendo parámetros como: número de veces que ganaron usando cierto personaje, la cantidad de daño infligido, cuantas veces es fue seleccionado un personaje, el número de partidas perdidas, entre otras. Esto con la finalidad de poder identificar la mejor opción al momento de escoger y así facilitar un poco la victoria, lo anterior mencionado es conocido como "meta".

El meta juego es un término proveniente de la lengua inglesa, "Most Efficient Tactic Available" o META hace referencia la táctica más eficiente disponible, esto es un concepto aplicado a todos los juegos, ya que, aunque el objetivo de los juego competitivos es mantener un balance, rara vez es posible, ya sea por las características de ciertos personajes o por las estrategias utilizadas por los jugadores los cuales beneficiaran de cierta manera a un personaje u otro, por lo que es necesario añadir balances o corregir las estadísticas para mantener un meta lo mas estable posible y así crear cierta armonía dentro del juego, esto con el fin de evitar frustraciones al momento de ir contra ciertos rivales o ciertos movimientos.

En el presente documento se muestra un dendograma realizado a partir de la base de datos "Valorant". Este proyecto nace a partir de la curiosidad de conocer cuáles de los personajes (agentes) del juego "Valorant" comparten características entre sí, ya sea en estadísticas dentro del propio juego o las obtenidas a través de los jugadores, por ejemplo, el número de veces que escogen a un agente, los resultados, puntajes, etc.

Fue realizado un dendograma donde se planea ejemplificar de manera más clara los distintos clústeres y agrupaciones, esto con el fin de poder identificar que agentes son los que comparten ciertas características entre sí,

Esta base contiene datos sobre el videojuego del mismo nombre, el cual es un juego de disparos táctico (Tactical Shooter).

Tema

Lanzado en 2020, el juego Valorant se convirtió en uno de los títulos más populares de este género, igualando o incluso superando la popularidad de otros como: "Counter Strike Global Offense" o "Tom Clancy's Rainbow Six 3".

Uno de los principales motivos por los cuales tuvo un rotundo éxito reside en los cambios en las mecánicas de este género, agregando efectos, personajes y habilidades más llamativas para los jugadores. De igual manera,

al ser desarrollado por la empresa RIOT GAMES (Creadores del juego "League Of Legends") contaba con un gran sequito de fanáticos los cuales decidieron comenzar y adentrarse dentro de su nuevo título.

Al ser un juego dentro de la categoria de "Tactical Shooter" el mundo competitivo ha estado presente desde sus inicios, por lo que, es necesario someter al juego a ajustes, actualizaciones y cambios regulares, esto con el fin de mantener una sensacion de frescura para los jugadores, de igual manera para cambiar el estilo y las cantidades de selección de los agentes los cuales son considerados "OP"(demasiado poderosos) por los jugadores.

Matriz de datos: dimensiones y variables

Esta matriz fue obtenida de la página "Kaggle".

La matriz de datos llamada "Valorant" cuenta con 17 variables y 18 objetos los cuales corresponden al nombre de los personajes dentro del juego.

Variables

- Agent: Nombre del personaje
- pick_rate: tasa de selección
- rounds: Rondas jugada
- ratings: Puntaje.
- ACS: Puntaje promedio de combate.
- K/D: Promedio asesinatos y muertes.
- ADR: Daño promedio por ronda.
- KPR: Asesinatos por rondas.
- DPR: Daño por ronda.
- APR: Cantidad promedio de asistencias por ronda.
- FBPR: Primer asesinato por ronda.
- FDPR: Primera muerte por ronda.
- HS: Porcentaje de tiros a la cabeza.
- FBSR: Primera Sangre exitosa.
- HUA: Si tiene un set de habilidades útiles.
- Nerf: Si ha sido nerfeado.
- CG: Sexo del personaje.

Exploración de las matriz

La matriz seleccionada fue obtenida en la página Kaggle, contiene datos sobre algunas estadisticas de los personajes del juego "Valorant", por ejemplo: nombre: tiros a la cabeza, sexo, si han sufrido nerfeos, etc.

```
library(readxl)
agentes <- agentsMAYBE <- read_excel("agentsMAYBE.xlsx")
head(agentsMAYBE)
```

```
## # A tibble: 6 x 17
##   agent  pick_rate rounds rating   ACS `K/D`   ADR   KPR   DPR   APR  FBPR  FDPR
##   <chr>      <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Jett       0.781 462467   1.08  233.  1.1   149.  0.82  0.74  0.16  0.19  0.16
## 2 Sova       0.76  450157   0.97  195.  0.98  134.  0.66  0.68  0.32  0.06  0.06
## 3 Astra      0.481 284853   0.98  187.  0.99  124.  0.68  0.68  0.32  0.07  0.07
## 4 Viper      0.427 253988   0.95  193.  0.97  128.  0.67  0.69  0.28  0.08  0.09
## 5 Killj~     0.343 202132   0.91  197.  1     134.  0.69  0.69  0.18  0.08  0.09
## 6 Chamb~     0.341 203821   1.01  215.  1.09  140.  0.77  0.71  0.15  0.13  0.11
## # ... with 5 more variables: `HS%` <dbl>, `FBSR%` <dbl>, HUA <dbl>, Nerf <dbl>,
```

```
## #   CG <chr>
```

Nota

La matriz de datos presenta tanto datos categóricos como númericos.

```
dim(agentes)
```

```
## [1] 18 17
```

Nota

Podemos observar que nuestra matriz cuenta con 17 variables y 18 objetos, los 18 objetos son los agentes dentro del juego.

```
str(agentes)
```

```
## tibble [18 x 17] (S3: tbl_df/tbl/data.frame)
##  $ agent    : chr [1:18] "Jett" "Sova" "Astra" "Viper" ...
##  $ pick_rate: num [1:18] 0.781 0.76 0.481 0.427 0.343 ...
##  $ rounds   : num [1:18] 462467 450157 284853 253988 202132 ...
##  $ rating   : num [1:18] 1.08 0.97 0.98 0.95 0.91 1.01 0.94 0.93 1.02 1.07 ...
##  $ ACS      : num [1:18] 233 195 187 193 197 ...
##  $ K/D      : num [1:18] 1.1 0.98 0.99 0.97 1 1.09 0.93 0.9 0.92 1.05 ...
##  $ ADR      : num [1:18] 149 134 124 128 134 ...
##  $ KPR      : num [1:18] 0.82 0.66 0.68 0.67 0.69 0.77 0.66 0.63 0.66 0.79 ...
##  $ DPR      : num [1:18] 0.74 0.68 0.68 0.69 0.69 0.71 0.71 0.7 0.72 0.75 ...
##  $ APR      : num [1:18] 0.16 0.32 0.32 0.28 0.18 0.15 0.29 0.33 0.43 0.23 ...
##  $ FBPR     : num [1:18] 0.19 0.06 0.07 0.08 0.08 0.13 0.08 0.07 0.08 0.15 ...
##  $ FDPR     : num [1:18] 0.16 0.06 0.07 0.09 0.09 0.11 0.1 0.09 0.1 0.14 ...
##  $ HS%      : num [1:18] 0.25 0.25 0.26 0.25 0.22 0.25 0.26 0.25 0.23 0.18 ...
##  $ FBSR%    : num [1:18] 0.54 0.49 0.48 0.46 0.48 0.55 0.45 0.45 0.46 0.5 ...
##  $ HUA      : num [1:18] 1 1 1 0 0 1 0 0 1 1 ...
##  $ Nerf     : num [1:18] 1 1 0 0 0 1 0 0 1 1 ...
##  $ CG       : chr [1:18] "f" "m" "f" "f" ...
```

Nota

Es posible apreciar que la mayoria de las variables en nuestra matriz son de cáracteR numerico.

```
anyNA(agentes)
```

```
## [1] FALSE
```

Nota

Por último, verificamos que no existan NA´s en nuestra matriz ya que esto podría ocasionarnos problemas.

Tratamiento de la matriz

Aparentemente en nuestra matriz existen datos “tibbles” en la variable “agent” los cuales generan conflicto al momento de realizar el dendograma, por lo que procedemos a crear una nueva variable llamada “personaje” con los nombres de los personajes.

```
personaje <- c("Astra","Breach", "Brimstone", "Chamber", "Cypher",               "Jett", "KAYO","Killjo
```

Renombramos nuestra matriz

```
AMM=as.data.frame(agentes)
```

Combinamos la nueva variable y nuestros datos

```
dfnombres <- cbind(AMM, personaje)
head(dfnombres)
```

```
##     agent pick_rate rounds rating   ACS  K/D   ADR  KPR  DPR  APR FBPR FDPR
## 1    Jett    0.7814 462467   1.08 232.8 1.10 148.7 0.82 0.74 0.16 0.19 0.16
## 2    Sova    0.7600 450157   0.97 194.6 0.98 133.5 0.66 0.68 0.32 0.06 0.06
## 3   Astra    0.4814 284853   0.98 186.6 0.99 123.8 0.68 0.68 0.32 0.07 0.07
## 4   Viper    0.4267 253988   0.95 193.1 0.97 128.4 0.67 0.69 0.28 0.08 0.09
## 5 Killjoy    0.3427 202132   0.91 197.4 1.00 134.3 0.69 0.69 0.18 0.08 0.09
## 6 Chamber    0.3414 203821   1.01 214.6 1.09 140.2 0.77 0.71 0.15 0.13 0.11
##    HS% FBSR% HUA Nerf CG personaje
## 1 0.25  0.54   1    1  f     Astra
## 2 0.25  0.49   1    1  m    Breach
## 3 0.26  0.48   1    0  f Brimstone
## 4 0.25  0.46   0    0  f   Chamber
## 5 0.22  0.48   0    0  f    Cypher
## 6 0.25  0.55   1    1  m      Jett
```

Metodología de análisis.

Dendograma

El dendograma es un diagrama comunmente con forma de arbol el cual presenta los clusters o grupos los cuales fueron creados al realizar conglomeraciones de nuestras observaciones, asi como sus niveles de similitud. Por lo general el nivel de similitud se mide en el eje vertical.

Este tipo de gráfico permite observar con mayor claridad las relaciones de nuestras agrupaciones entre los datos.

De igual manera, podriamos decir que el dendograma es una ilustración de las agrupaciones las cuales se originaron a partir de las técnicas de un algoritmo de clustering jerárquico.

Clustering CLARA

Una de las limitaciones del método K-medoids-clustering es el algoritmo necesita demasiada memoria RAM, lo que impide que se pueda aplicar cuando el set de datos contiene grandes cantidades de observaciones. CLARA (Clustering Large Applications) es un método el cual utiliza algunas ideas del procedimiento de K-medoids con el resampling para que ser aplicado.

En vez de intentar encontrar los medoids utilizando todos nuestros datos, el metodo CLARA seleccionará una muestra aleatoria de un tamaño determinado y utilizará el algoritmo de PAM (K-medoids) para encontrar los clusters óptimos acorde a esa muestra. Utilizando esos medoids se agrupan las observaciones de todo el set de datos. La calidad de los medoids resultantes se cuantifica con la suma total de las distancias entre cada observación del set de datos y su correspondiente medoid (suma total de distancias intra-clusters). CLARA repite este proceso un número predeterminado de veces con el objetivo de reducir el bias de muestreo. Por último, se seleccionan como clusters finales los obtenidos con aquellos medoids que han conseguido menor suma total de distancias. A continuación, se describen los pasos del algoritmo CLARA.

Se divide aleatoriamente el set de datos en n partes de igual tamaño, donde n es un valor que determina el analista.

Para cada una de las n partes:

- Aplicar el algoritmo PAM e identificar cuáles son los k medoids.
- Utilizando los medoids del paso anterior agrupar todas las observaciones del set de datos.
- Calcular la suma total de las distancias entre cada observación del set de datos y su correspondiente medoid (suma total de distancias intra-clusters).
- Seleccionar como clustering final aquel que ha conseguido menor suma total de distancias intra-clusters en el paso 2.3.

Resultados

Cálculo de la matriz de distancia de Mahalonobis

```
dist.AMM<-dist(dfnombres[,2:14])
```

Seleccionamos las variables númericas

```
w <- round(as.matrix(dist.AMM)[1:13, 1:18],3)
head(w)
```

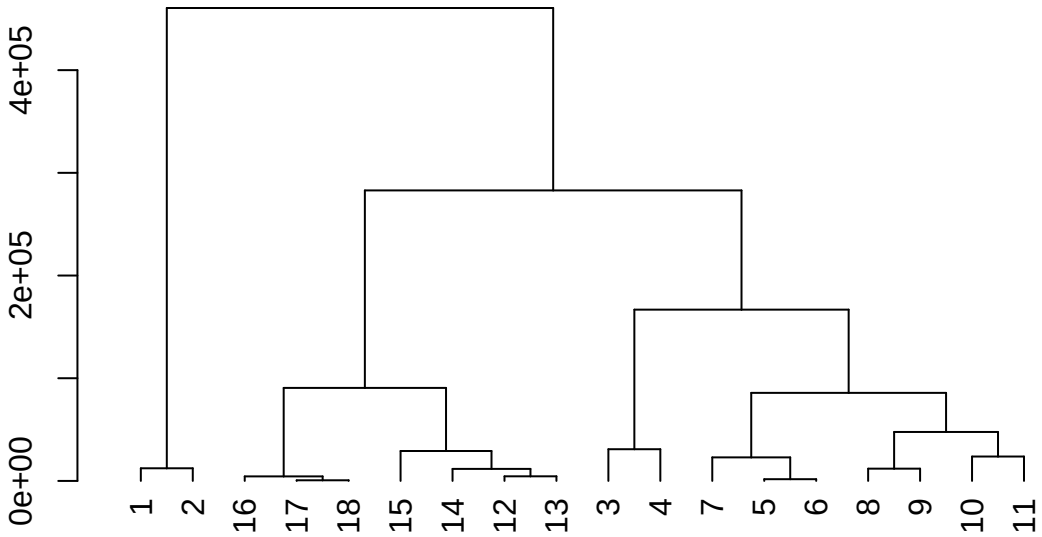
```
##           1         2         3         4          5          6         7
## 1      0.00  12310.07 177614.01 208479.01 260335.003 258646.001 281554.01
## 2  12310.07      0.00 165304.00 196169.00 248025.000 246336.001 269244.00
## 3 177614.01 165304.00      0.00  30865.00  82721.001  81032.006 103940.00
## 4 208479.01 196169.00  30865.00      0.00  51856.001  50167.006  73075.00
## 5 260335.00 248025.00  82721.00  51856.00      0.000   1689.098  21219.01
## 6 258646.00 246336.00  81032.01  50167.01   1689.098      0.000  22908.02
##           8         9        10        11     12     13     14     15     16
## 1 296723.01 308656.00 320642.00 344390.01 369913 374399 381612 399037 456079
## 2 284413.00 296346.00 308332.00 332080.00 357603 362089 369302 386727 443769
## 3 119109.00 131042.00 143028.01 166776.00 192299 196785 203998 221423 278465
## 4  88244.00 100177.00 112163.01 135911.00 161434 165920 173133 190558 247600
## 5  36388.01  48321.00  60307.02  84055.01 109578 114064 121277 138702 195744
## 6  38077.02  50010.01  61996.01  85744.01 111267 115753 122966 140391 197433
##       17     18
## 1 459874 460489
## 2 447564 448179
## 3 282260 282875
## 4 251395 252010
## 5 199539 200154
## 6 201228 201843
```

Calculo del dendrograma

```
dend.AMM<-as.dendrogram(hclust(dist.AMM))
```

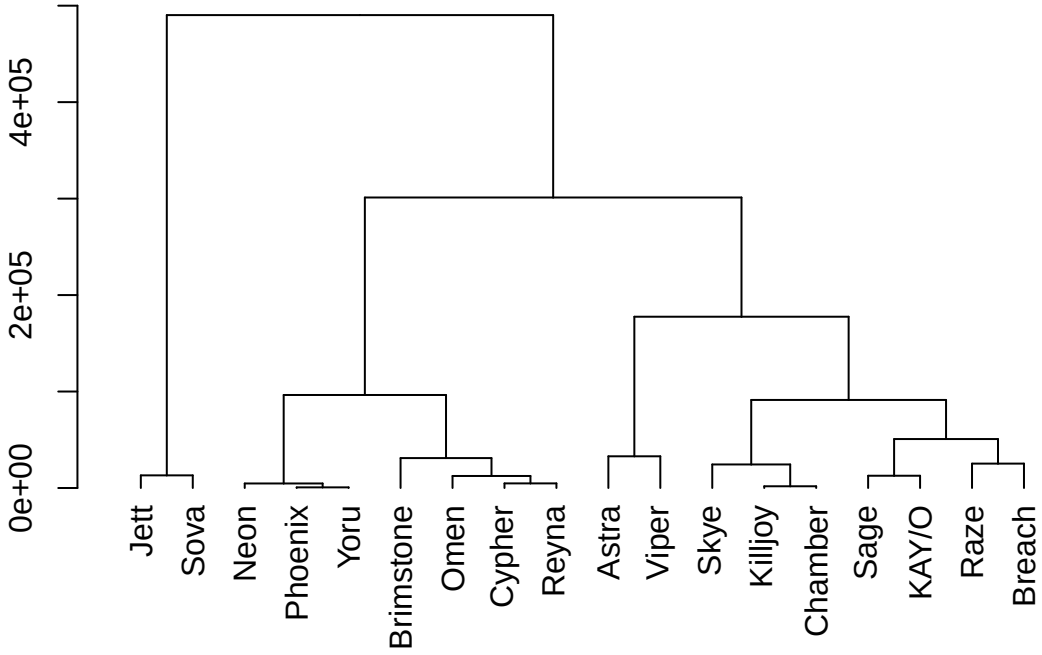
Generacion del dendrograma

```
plot(dend.AMM)
```




```
AMM.nombres=dfnombres
rownames(AMM.nombres)= AMM.nombres$agent
AMM.nombres=AMM.nombres[,-1]
```

```
plot(as.dendrogram(hclust(dist(AMM.nombres))))
```



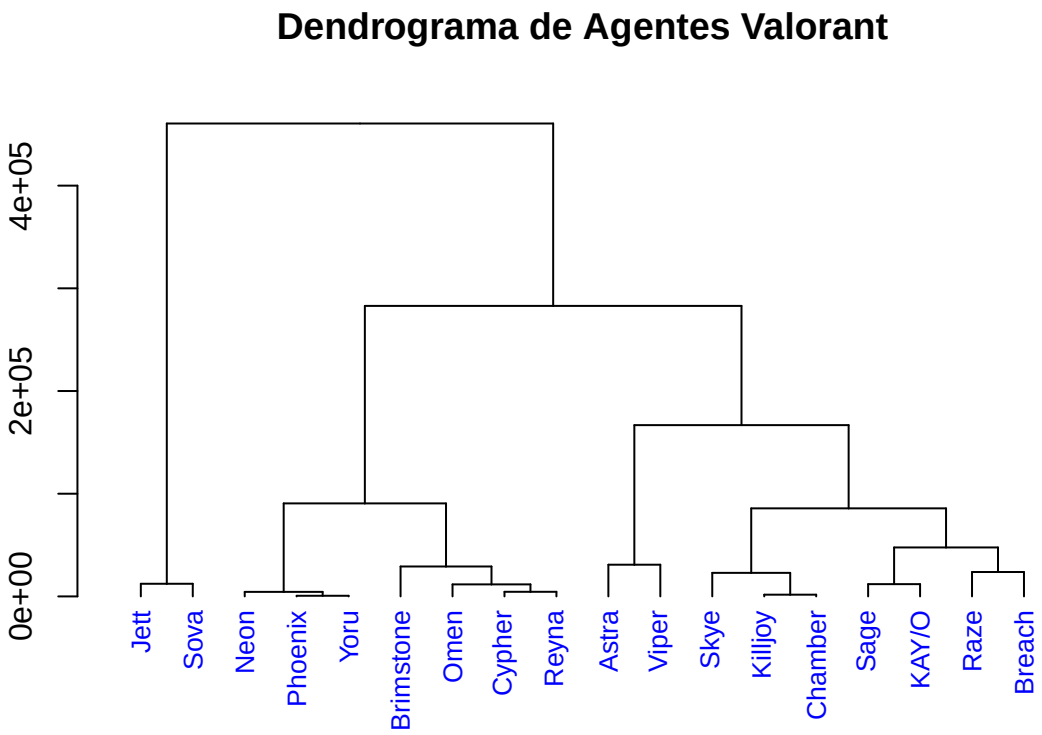
Modificar el dendrograma

```
install.packages("dendextend")
library(dendextend)
```

```
L=labels(dend.AMM)

labels(dend.AMM)= dfnombres$agent[L]
```

```
dend.AMM %>%
  set(what="labels_col", "blue") %>% #Colores etiqueta
  set(what="labels_cex", 0.8) %>%
  plot(main="Dendrograma de Agentes Valorant")
```



Interpretación gráfico 1

En este gráfico es posible apreciar como todos los agentes fueron dividas principalmente en 3 clústeres, el primero es el que más destaca debido que solo cuenta son 2 agentes, mientras que los otros dos clústeres cuentan con y 7 y 9 respectivamente.

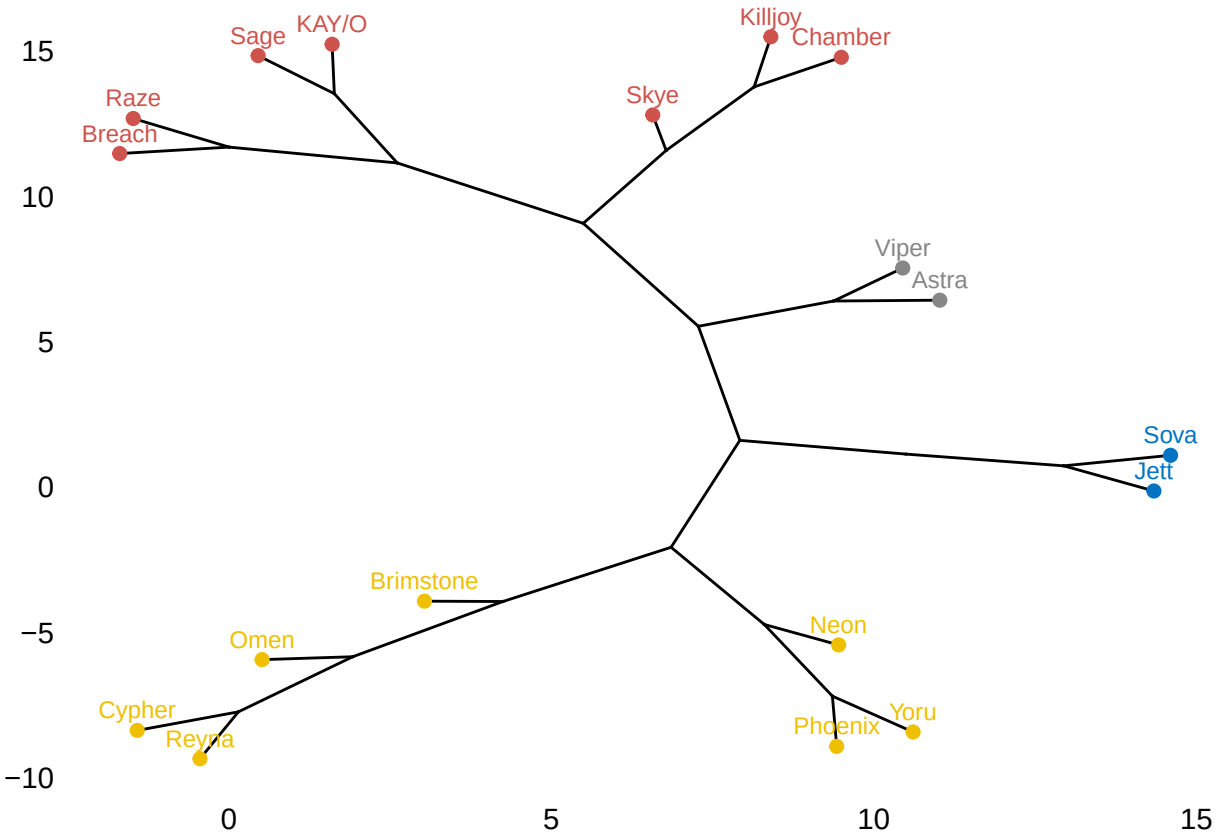
Dendograma diseño filogénicos.

```
install.packages("igraph")
install.packages("stats")
install.packages("factoextra")
install.packages("scales")
install.packages("ggsci")

library(igraph)
library(stats)
library(factoextra)
library(scales)
library(ggsci)
```

```
Phylo = fviz_dend(dend.AMM, cex = 0.8, lwd = 0.8, k = 4,
                  rect = TRUE,
                  k_colors = "jco",
                  rect_border = "jco",
```

```
                  rect_fill = TRUE,
                  type = "phylogenic")
Phylo
```



Interpretación gráfico 2

En este gráfico es posible apreciar las divisiones de los clusters de una manera mas clara, ya que podemos ver a los agentes dividos en distintas ramas las cuales representan las catacteristicas que comparten, mientrsa mas cercanos esten significará que comparten mas cualidades entre sí.

Clustering CLARA

Creación de matriz con las variables seleccionadas.

```
datosclara2 <- cbind(AMM$pick_rate, AMM$rating, AMM$KPR, AMM$DPR, AMM$APR, AMM$FBPR, AMM$`HS%`)
colnames(datosclara2) <- c("x", "y", "z", "w","a","b", "c")
head(datosclara2)
```

```
##           x    y    z    w    a    b    c
## [1,] 0.7814 1.08 0.82 0.74 0.16 0.19 0.25
## [2,] 0.7600 0.97 0.66 0.68 0.32 0.06 0.25
## [3,] 0.4814 0.98 0.68 0.68 0.32 0.07 0.26
## [4,] 0.4267 0.95 0.67 0.69 0.28 0.08 0.25
## [5,] 0.3427 0.91 0.69 0.69 0.18 0.08 0.22
## [6,] 0.3414 1.01 0.77 0.71 0.15 0.13 0.25
```

```r
install.packages("cluster")
install.packages("factoextra")

library(cluster)
library(factoextra)
```

```r
clara_clusters <- clara(x = datosclara2, , k = 2, metric = "manhattan", stand = TRUE,
                        samples = 17, pamLike = TRUE)
clara_clusters
```

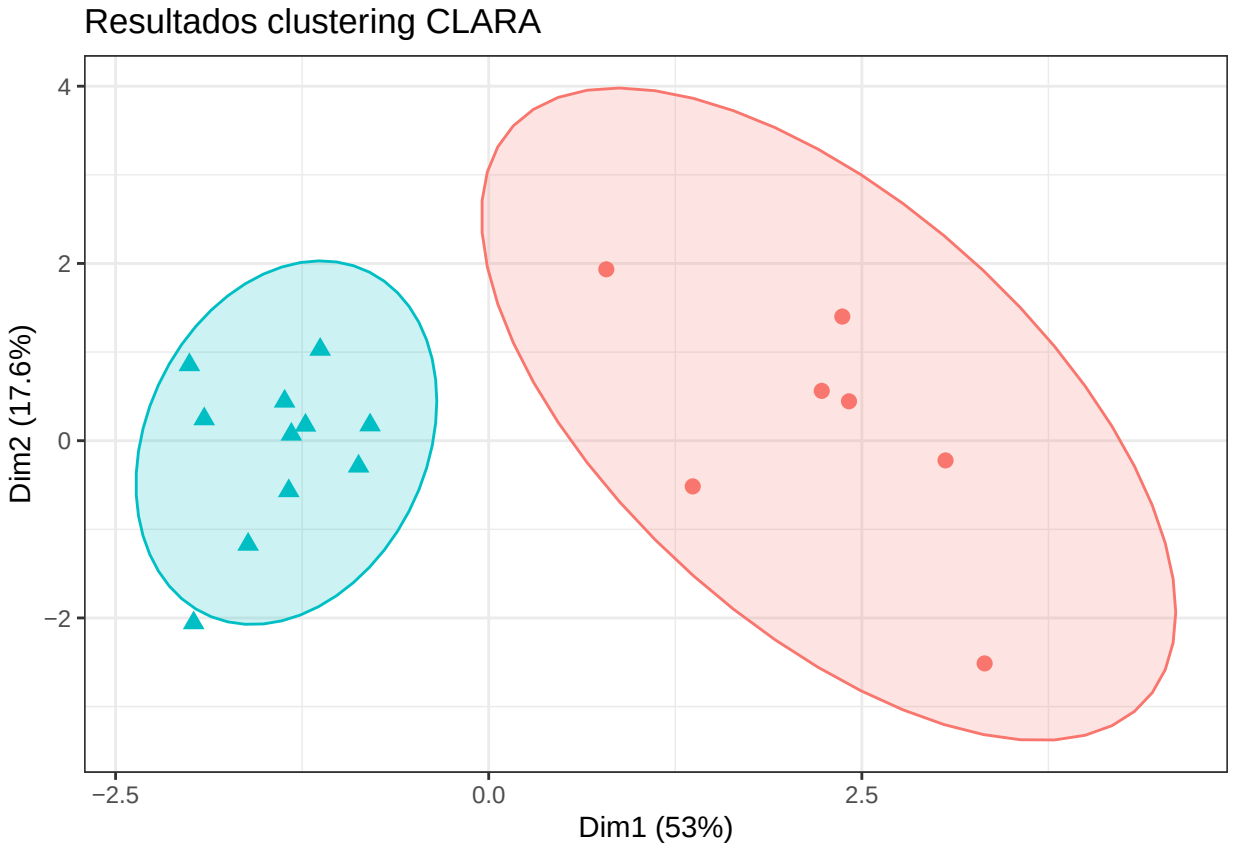
```
## Call:     clara(x = datosclara2, k = 2, metric = "manhattan", stand = TRUE,      samples = 17, pamLi
## Medoids:
##           x    y    z    w    a    b    c
## [1,] 0.1512 1.04 0.79 0.76 0.18 0.13 0.26
## [2,] 0.1358 0.97 0.66 0.70 0.33 0.07 0.25
## Objective function:   4.514326
## Clustering vector:    int [1:18] 1 2 2 2 2 1 2 2 2 1 2 2 1 2 2 1 1 1
## Cluster sizes:            7 11
## Best sample:
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
##
## Available components:
##  [1] "sample"     "medoids"    "i.med"      "clustering" "objective"
##  [6] "clusinfo"   "diss"       "call"       "silinfo"    "data"
```

```r
summary(clara_clusters)
```

```
## Object of class 'clara' from call:
##  clara(x = datosclara2, k = 2, metric = "manhattan", stand = TRUE,      samples = 17, pamLike = TRUE)
## Medoids:
##           x    y    z    w    a    b    c
## [1,] 0.1512 1.04 0.79 0.76 0.18 0.13 0.26
## [2,] 0.1358 0.97 0.66 0.70 0.33 0.07 0.25
## Objective function:    4.514326
## Numerical information per cluster:
##      size max_diss  av_diss isolation
## [1,]    7 8.706206 5.446514 0.7966195
## [2,]   11 7.767962 3.921116 0.7107700
## Average silhouette width per cluster:
## [1] 0.3297513 0.5551534
## Average silhouette width of best sample: 0.467497
##
## Best sample:
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
## Clustering vector:
##  [1] 1 2 2 2 2 1 2 2 2 1 2 2 1 2 2 1 1 1
##
## Silhouette plot information for best sample:
##    cluster neighbor sil_width
## 13       1        2 0.4748668
## 17       1        2 0.4649646
## 16       1        2 0.4013530
## 10       1        2 0.3846772
## 1        1        2 0.3581441
## 6        1        2 0.1498798
```

```
## 18       1        2 0.0743735
## 8        2        1 0.6675870
## 14       2        1 0.6394560
## 4        2        1 0.6298498
## 11       2        1 0.6151020
## 7        2        1 0.6022872
## 3        2        1 0.5845765
## 2        2        1 0.5603041
## 15       2        1 0.5130136
## 12       2        1 0.4737654
## 9        2        1 0.4264315
## 5        2        1 0.3943139
##
## 153 dissimilarities, summarized :
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.059   6.429   8.892   9.539  12.823  19.170
## Metric :  manhattan
## Number of objects : 18
##
## Available components:
##  [1] "sample"     "medoids"    "i.med"      "clustering" "objective"
##  [6] "clusinfo"   "diss"       "call"       "silinfo"    "data"
```

```
fviz_cluster(object = clara_clusters, ellipse.type = "t", geom = "point",
             pointsize = 2.5) +
  theme_bw() +
  labs(title = "Resultados clustering CLARA") +
  theme(legend.position = "none")
```



Interpretación gráfico 3

En este gráfico es posible apreciar la division o clasificación de grupos a traves del método CLARA el cual separa los k-medoides.

Conclusiones

En nuestro gráfico 1 es posible apreciar como tenemos 3 clústeres principales, en el primero encontramos a los agentes Jett y Sova (Grupo A), en el segundo tenemos a Neon, Phoenix, Yoru, Brimstone, Omen, Cypher Y Reyna (Grupo B) y en el tercero tenemos a Astra, Viper, Skye, Killjoy,Chamber,Sage, KAY/O y Breach (Grupo C).

Es posible notar que tanto Jett como Sova comparten características similares entre ellos mismos, pero distintas al resto de los agentes, algunas de estas características son la tasa de selección, las rondas jugadas, los tiros a la cabeza, habilidades útiles, y los nerfeos. Al observar estas variables es posible notar y concluir que, por lo general los jugadores obtienen resultados similares al momento de utilizar a estos agentes.

Hablando un poco más el subgrupo de Neon, Phoenix, Yoru (B1) tenemos que algunas de las características compartidas son: el puntaje, el puntaje promedio de combate, el K/d, el ADR entre otros. Es posible observar como el gráfico agrupa a los agentes en grupo más cercanos, por lo que es posible concluir que la mayoría de los agentes en este subgrupo comparten sus características, y aquellas que características los distinguen son mucho menores a aquellas que los relacionan.

Por último, en el subgrupo c tenemos a los agentes Astra, Viper, Skye, Killjoy, Chamber, Sage, KAYO, Raze y Breach. En este grupo encontramos que algunas de las características son similares al grupo anterior, sin embargo, el distintivo se encuentra en las variables FBPR y FBSR%, estas presentan valores más altos en este grupo de personajes lo cual los posiciona en este Clúster.

En conclusión, una vez observado el dendograma, podemos concluir que la mayoría de los agentes del juego comparten características entre sí, a excepción de los agentes Jett y Sova, los cuales podemos decir en base a la matriz, son los favoritos de los jugadores. De igual manera son estos agentes los cuales cuentan con las estadísticas más elevadas, lo cual significa que son los personajes más en "META"

El gráfico 2 es una variante del dendograma tradicional, es llamado dendograma filogénico, en este los clústeres se encuentran representados de una manera mucho más clara, ya que permite observar una separación entre los distintos agentes, ya que, para cada grupo el cual comparte ciertas características se genera una "rama".

En este caso, mientras más cerca de las ramificaciones se encuentren, significará que comparten más características, por ejemplo, los agentes Cypher y Reyna comparten las caracteristicas tasa de selección, rondas jugadas, K/D, KPR, DPR, FBSR y HUA, mientras que los agentes Omen y Brimstone comparten algunas de ellas pero no las mismas, por lo que están graficados un poco alejados de ellos, sin embargo son agrupados en la misma rama.

A partir de la técnica de clustering CLARA podemos observar como usando muestras aleatorias de nuestras variables se generan agrupaciones a partir de los datos los cuales comparten caractetisticas.

Es posible observar en el grupo azul como un dato no esta bien clasificado, aunque no es de gran importancia ya que la mayoria de nuestros datos se encuentran dentro de los grupos cabe mencionar que esto puede ser debido a que es un valor mayor a los demas, como puede ser el caso en la variable "Rondas".

A partir de la técnica de clustering CLARA podemos observar cómo usando muestras aleatorias de nuestras variables se generan agrupaciones a partir de los datos los cuales comparten características.

Es posible observar en el grupo azul como un dato no está bien clasificado, aunque no es de gran importancia ya que la mayoría de nuestros datos se encuentran dentro de los grupos cabe mencionar que esto puede ser debido a que es un valor mayor a los demás, como puede ser el caso en la variable "Rondas".

Referencias

- Amat, J. (2017, septiembre). Clustering y heatmaps: aprendizaje no supervisado. RPubs. Recuperado 26 de mayo de 2022, de https://rpubs.com/Joaquin_AR/310338
- Csardi G, Nepusz T: The igraph software package for complex network research, InterJournal, Complex Systems 1695. 2006. https://igraph.org
- Isaac, J. (2021, 22 abril). Cluster jerarquico en R. RPubs. Recuperado 26 de mayo de 2022, de https://rpubs.com/jaimeisaacp/760355
- Kassambara A, Mundt F (2020). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.7, http://www.sthda.com/english/rpkgs/factoextra.
- Kaufman, Leonard, and Peter Rousseeuw. 1990. Finding Groups in Data: An Introduction to Cluster Analysis
- Maechler, M., Rousseeuw, P., Struyf, A., Hubert, M., Hornik, K.(2022). cluster: Cluster Analysis Basics and Extensions. R package version 2.1.3.
- R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.
- Tal Galili (2015). dendextend: an R package for visualizing, adjusting, and comparing trees of hierarchical clustering. Bioinformatics. DOI: 10.1093/bioinformatics/btv428
- Wickham H, Seidel D (2022). *scales: Scale Functions for Visualization*. https://scales.r-lib.org, https://github.com/r-lib/scales.

- Wickham H, Bryan J (2022). *readxl: Read Excel Files*. https://readxl.tidyverse.org, https://github.com/tidyverse/readxl.
- Xiao N (2018). *ggsci: Scientific Journal and Sci-Fi Themed Color Palettes for 'ggplot2'*. https://nanx.me/ggsci/, https://github.com/road2stat/ggsci.