Lab 2 - Aprendizaje No Supervisado

2026-02-23

Lab 2 - Aprendizaje no supervisado

Clustering

Preprocesamiento

s

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.2.0     v readr     2.1.6
## v forcats   1.0.1     v stringr   1.6.0
## v ggplot2   4.0.2     v tibble    3.3.1
## v lubridate 1.9.5     v tidyr     1.3.2
## v purrr     1.2.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
##
## Package `clustertend` is deprecated.  Use package `hopkins` instead.
##
##
## ---------------------
## Welcome to dendextend version 1.19.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
##
##
##
## Adjuntando el paquete: 'dendextend'
##
##
## The following object is masked from 'package:stats':
```

```
##
##     cutree

## 'data.frame':    19883 obs. of  28 variables:
##  $ id                         : int  1627085 1626914 1626898 1626808 1626678 1626234 1626010 1625551 1
##  $ budget                     : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ genres                     : chr  "Drama|Crime" "Animation" "Animation" "Thriller|Mystery|Documentar
##  $ homePage                   : chr  "" "" "" "" ...
##  $ productionCompany          : chr  "" "" "" "" ...
##  $ productionCompanyCountry   : chr  "" "" "" "" ...
##  $ productionCountry          : chr  "" "" "" "" ...
##  $ revenue                    : num  0 0 0 0 0 1 0 0 0 0 ...
##  $ runtime                    : int  95 3 2 5 12 14 39 90 96 106 ...
##  $ video                      : logi  FALSE FALSE FALSE FALSE FALSE FALSE ...
##  $ director                   : chr  "Javad Hakami" "Kimmy Gatewood" "Kimmy Gatewood" "Felipe Roldán"
##  $ actors                     : chr  "Mohsen Ghasabian|Aida Mahiani|Mehran Ghafourian|Payam Ahmadinia|M
##  $ actorsPopularity           : chr  "0.3453|0.1664|0.9684|0.3437|0.3713|0.2437|0.2796|0.2639" "0|0.007
##  $ actorsCharacter            : chr  "|||||||" "Prince Charming|Evil Stepmother|Fairy Godmother|Cinderel
##  $ originalTitle              : chr    " "  "Cinderella" "Aladdin" "EL ANILLO Y EL DECK" ...
##  $ title                      : chr  "Immersed" "Cinderella" "Aladdin" "THE RING AND THE DECK" ...
##  $ originalLanguage           : chr  "fa" "en" "en" "es" ...
##  $ popularity                 : num  0.0357 0.0357 0.0214 0.0429 0.0379 ...
##  $ releaseDate                : chr  "2026-02-01" "2026-02-01" "2026-02-01" "2026-02-01" ...
##  $ voteAvg                    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ voteCount                  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ genresAmount               : int  2 1 1 3 1 1 1 1 3 1 ...
##  $ productionCoAmount         : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ productionCountriesAmount: int  0 0 0 0 0 0 0 1 1 0 ...
##  $ actorsAmount               : int  8 4 3 7 3 3 5 4 5 5 ...
##  $ castWomenAmount            : int  2 0 0 0 0 0 0 3 1 2 ...
##  $ castMenAmount              : int  5 0 0 0 0 0 3 0 3 3 ...
##  $ releaseYear                : int  2026 2026 2026 2026 2026 2026 2026 2026 2026 2026 ...

##        id              budget             genres            homePage
##  Min.   :      5   Min.   :        0   Length:19883       Length:19883
##  1st Qu.: 146220   1st Qu.:        0   Class :character   Class :character
##  Median : 869623   Median :        0   Mode  :character   Mode  :character
##  Mean   : 902240   Mean   :  9413280
##  3rd Qu.:1589603   3rd Qu.:  1000000
##  Max.   :1627166   Max.   :380000000
##
##  productionCompany  productionCompanyCountry productionCountry
##  Length:19883       Length:19883             Length:19883
##  Class :character   Class :character         Class :character
##  Mode  :character   Mode  :character         Mode  :character
##
##
##
##
##     revenue             runtime         video           director
##  Min.   :0.000e+00   Min.   :  0.00   Mode :logical   Length:19883
##  1st Qu.:0.000e+00   1st Qu.: 10.00   FALSE:19313     Class :character
##  Median :0.000e+00   Median : 86.00   TRUE :84        Mode  :character
##  Mean   :2.879e+07   Mean   : 66.09   NA's :486
```

```
##  3rd Qu.:3.306e+05   3rd Qu.:103.00
##  Max.   :2.847e+09   Max.   :750.00
##
##     actors          actorsPopularity   actorsCharacter    originalTitle
##  Length:19883       Length:19883       Length:19883       Length:19883
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##     title           originalLanguage     popularity        releaseDate
##  Length:19883       Length:19883       Min.   :0.000e+00   Length:19883
##  Class :character   Class :character   1st Qu.:5.460e-02   Class :character
##  Mode  :character   Mode  :character   Median :8.502e+00   Mode  :character
##                                        Mean   :2.625e+01
##                                        3rd Qu.:2.224e+01
##                                        Max.   :1.147e+04
##
##     voteAvg         voteCount        genresAmount    productionCoAmount
##  Min.   : 0.000   Min.   :    0.0   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 0.000   1st Qu.:    0.0   1st Qu.: 1.000   1st Qu.: 0.000
##  Median : 5.400   Median :    6.0   Median : 2.000   Median : 1.000
##  Mean   : 3.837   Mean   :  675.9   Mean   : 1.949   Mean   : 1.973
##  3rd Qu.: 6.800   3rd Qu.:  423.0   3rd Qu.: 3.000   3rd Qu.: 3.000
##  Max.   :10.000   Max.   :30788.0   Max.   :16.000   Max.   :89.000
##
##  productionCountriesAmount  actorsAmount    castWomenAmount  castMenAmount
##  Min.   :  0.00             Min.   :     0   Min.   :     0   Min.   :     0
##  1st Qu.:  1.00             1st Qu.:     3   1st Qu.:     0   1st Qu.:     0
##  Median :  1.00             Median :     9   Median :     2   Median :     3
##  Mean   :  1.23             Mean   :  1082   Mean   :  3517   Mean   :  8224
##  3rd Qu.:  1.00             3rd Qu.:    21   3rd Qu.:     6   3rd Qu.:    12
##  Max.   :155.00             Max.   :919590   Max.   :922162   Max.   :922017
##                                              NA's   :37       NA's   :162
##   releaseYear
##  Min.   :1902
##  1st Qu.:2013
##  Median :2021
##  Mean   :2017
##  3rd Qu.:2025
##  Max.   :2026
##  NA's   :2
```

Selección de variables Eliminamos variables que no aportan a la formación de grupos:

id, originalTitle, title, homePage, actorsCharacter, releaseDate, director, productionCompany, actors

Variables textuales no numéricas

Trabajaremos con variables numéricas relevantes para comportamiento y desempeño:

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(everything(), ~as.numeric(as.character(.)))`.
## Caused by warning:
```

```
## ! NAs introducidos por coerción
```

Tendencia al Agrupamiento

```
set.seed(123)
hopkins_stat <- hopkins(movies_scaled, n = nrow(movies_scaled)-1)
```

Estadistico de Hopkins

```
## Warning in hopkins(movies_scaled, n = nrow(movies_scaled) - 1): Package
## `clustertend` is deprecated.  Use package `hopkins` instead.
```

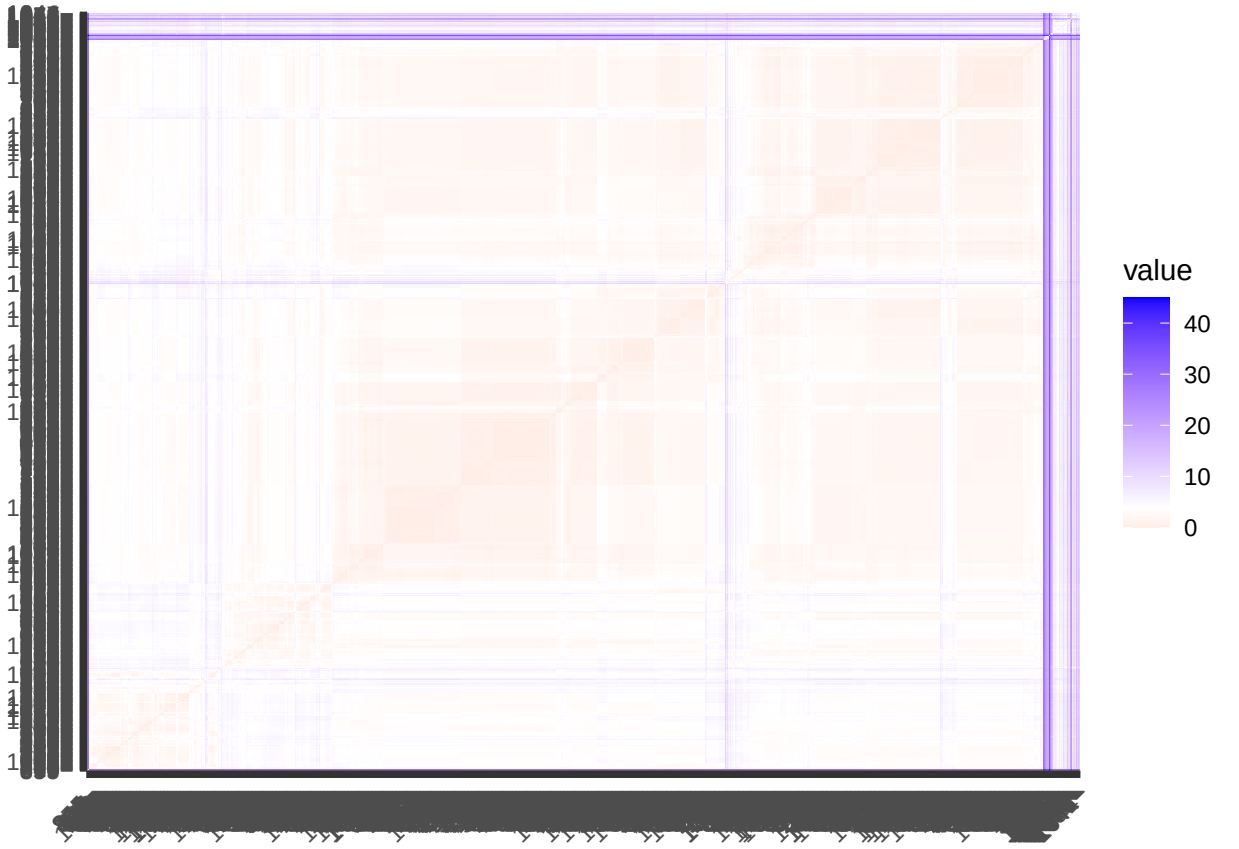
```
hopkins_stat
```

```
## $H
## [1] 0.02059509
```

```
fviz_dist(dist(movies_scaled))
```

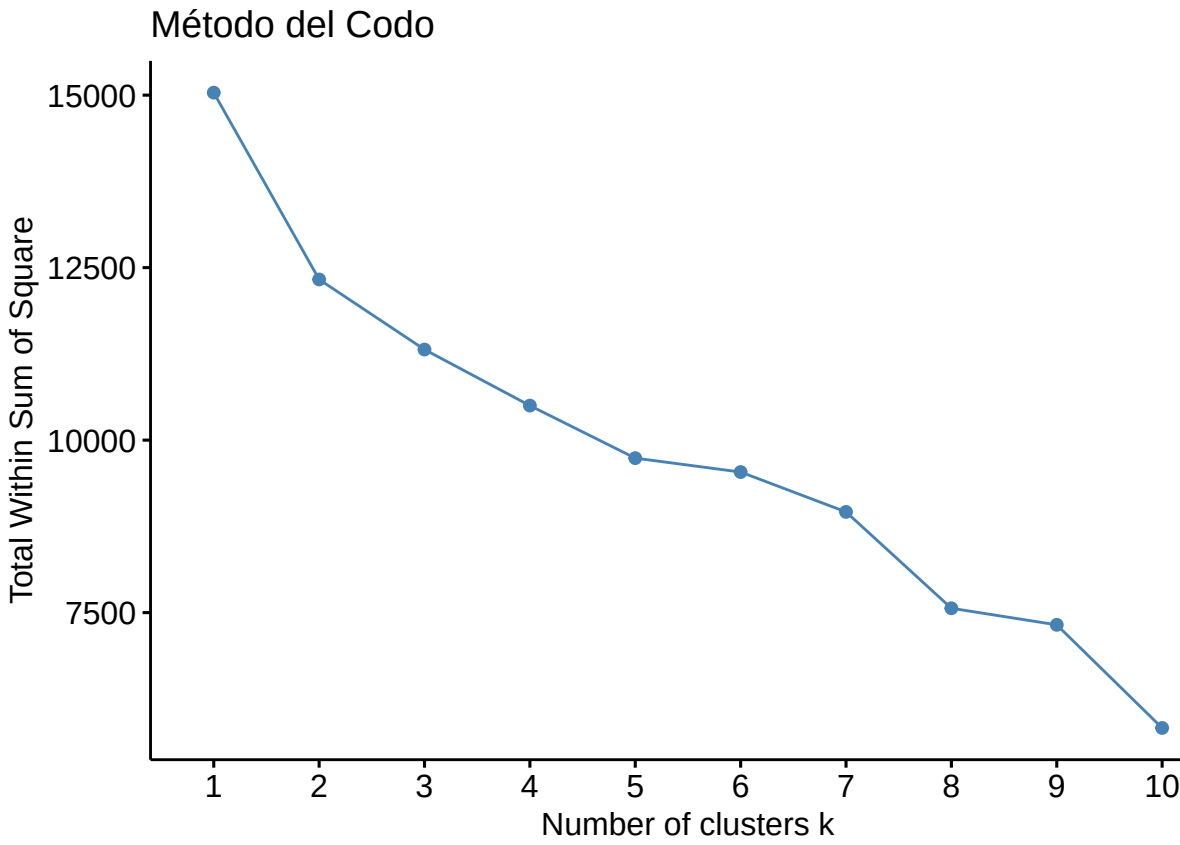
VAT

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once per session.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



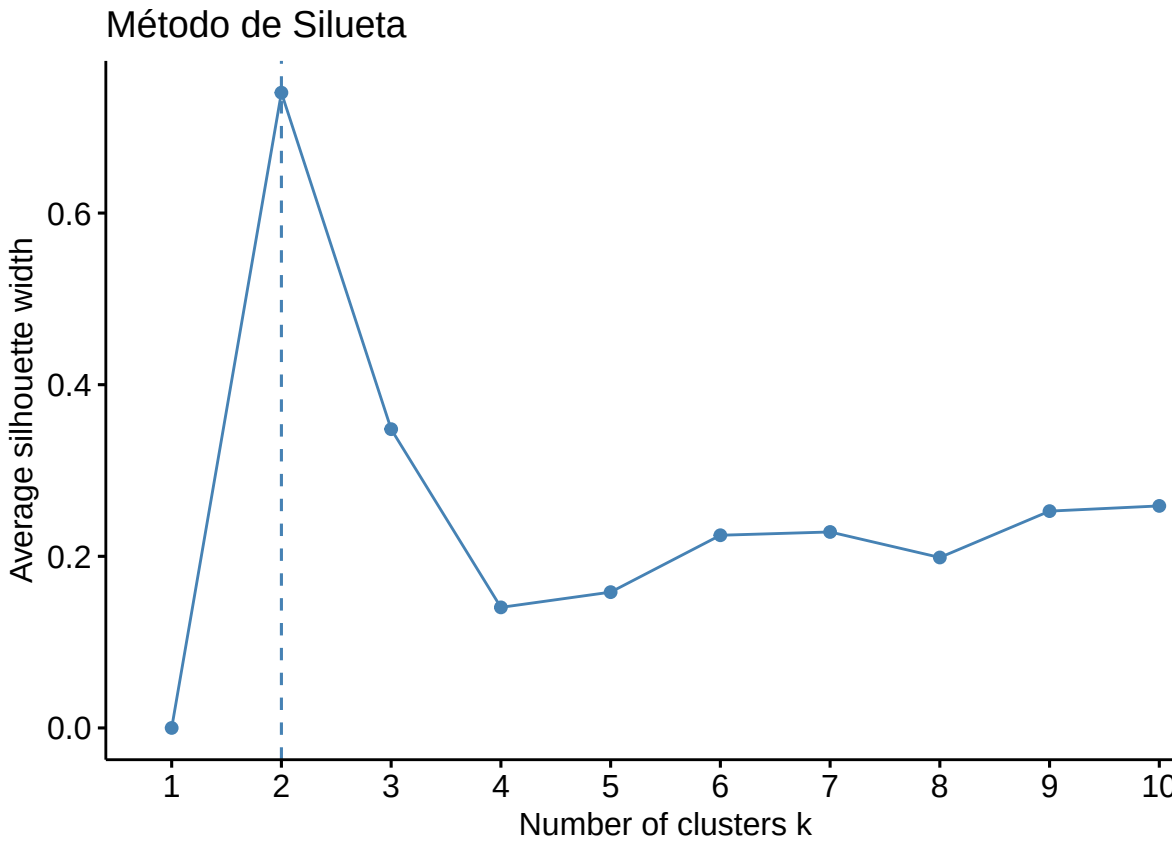
Numero Optimo de Clusters

```
fviz_nbclust(movies_scaled, kmeans, method = "wss") +
  ggtitle("Método del Codo")
```



Metodo del Codo

```
fviz_nbclust(movies_scaled, kmeans, method = "silhouette") +
  ggtitle("Método de Silueta")
```

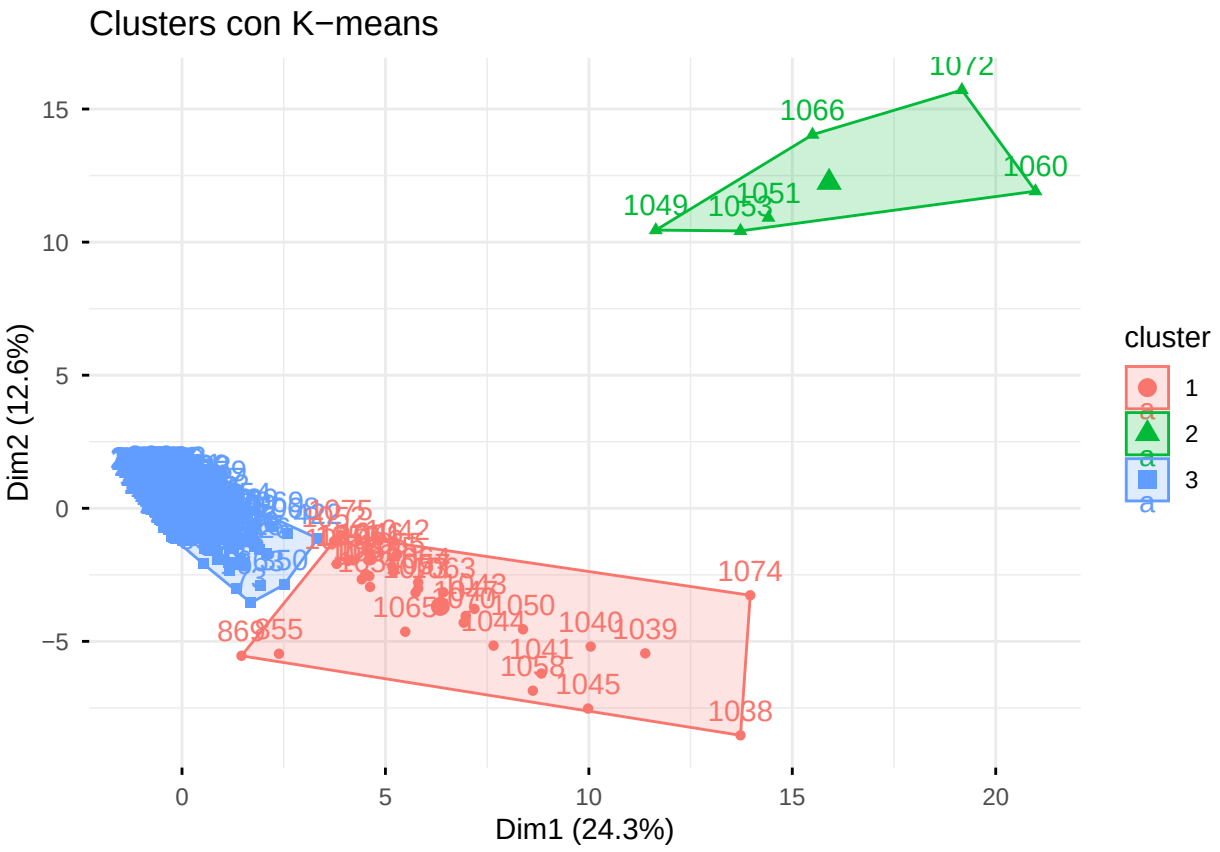


Metodo de silueta

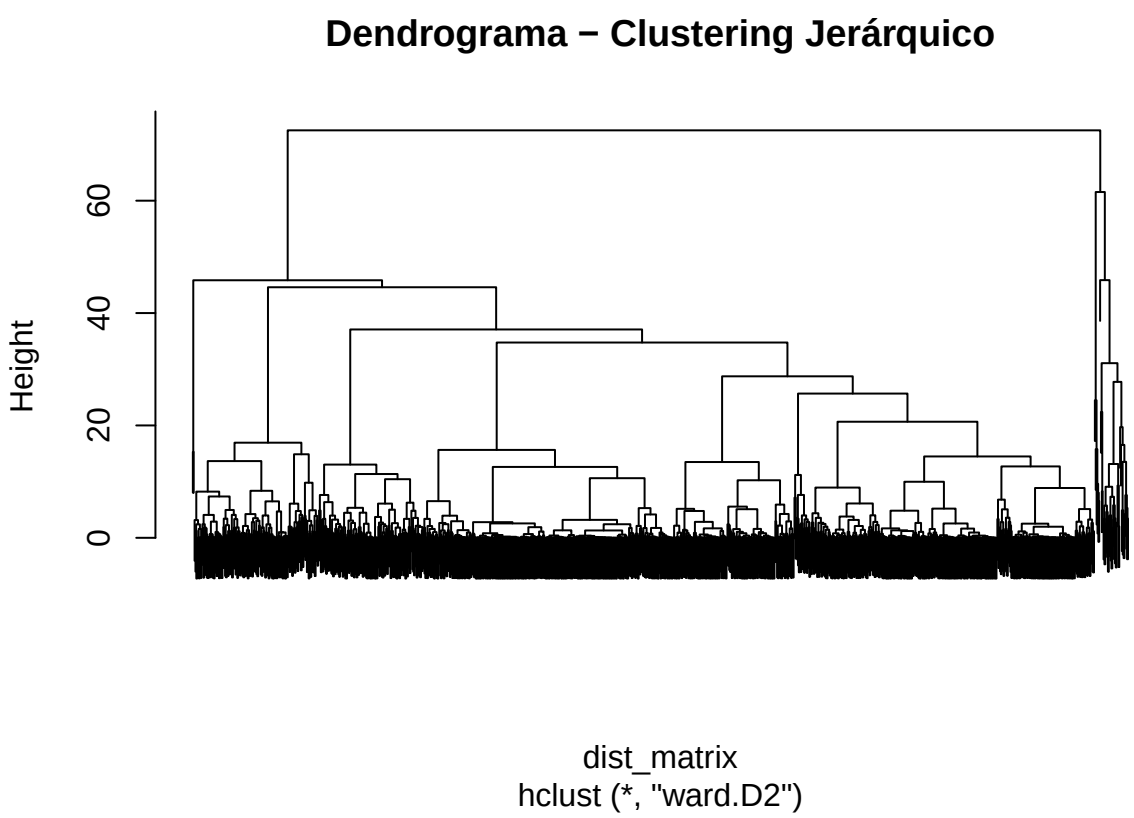
1.4 Aplicación de K-medias y Clustering Jerárquico

K-medias

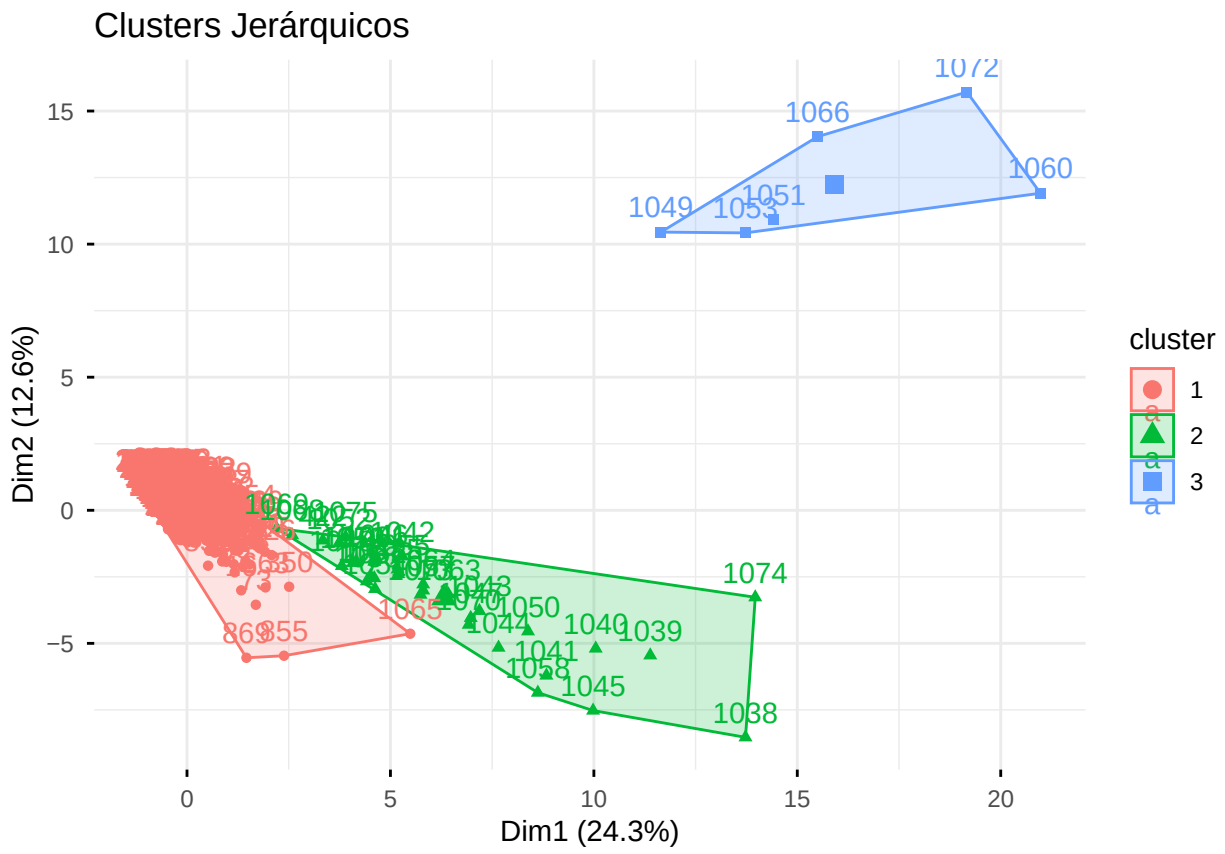
```
set.seed(123)

k_opt <- 3  # Ajustar según los métodos del codo y silueta

kmeans_model <- kmeans(movies_scaled, centers = k_opt, nstart = 25)

# Agregar cluster al dataset
movies_clean$cluster_kmeans <- as.factor(kmeans_model$cluster)

# Visualización
fviz_cluster(kmeans_model, data = movies_scaled,
             ellipse.type = "convex",
             ggtheme = theme_minimal(),
             main = "Clusters con K-means")
```




```r
# Matriz de distancias
dist_matrix <- dist(movies_scaled)

# Método de Ward
hc_model <- hclust(dist_matrix, method = "ward.D2")

# Dendrograma
plot(hc_model, labels = FALSE, main = "Dendrograma - Clustering Jerárquico")
```




```r
# Cortar el árbol en k clusters
hc_clusters <- cutree(hc_model, k = k_opt)

movies_clean$cluster_hc <- as.factor(hc_clusters)

# Visualización
fviz_cluster(list(data = movies_scaled, cluster = hc_clusters),
             ellipse.type = "convex",
             ggtheme = theme_minimal(),
             main = "Clusters Jerárquicos")
```

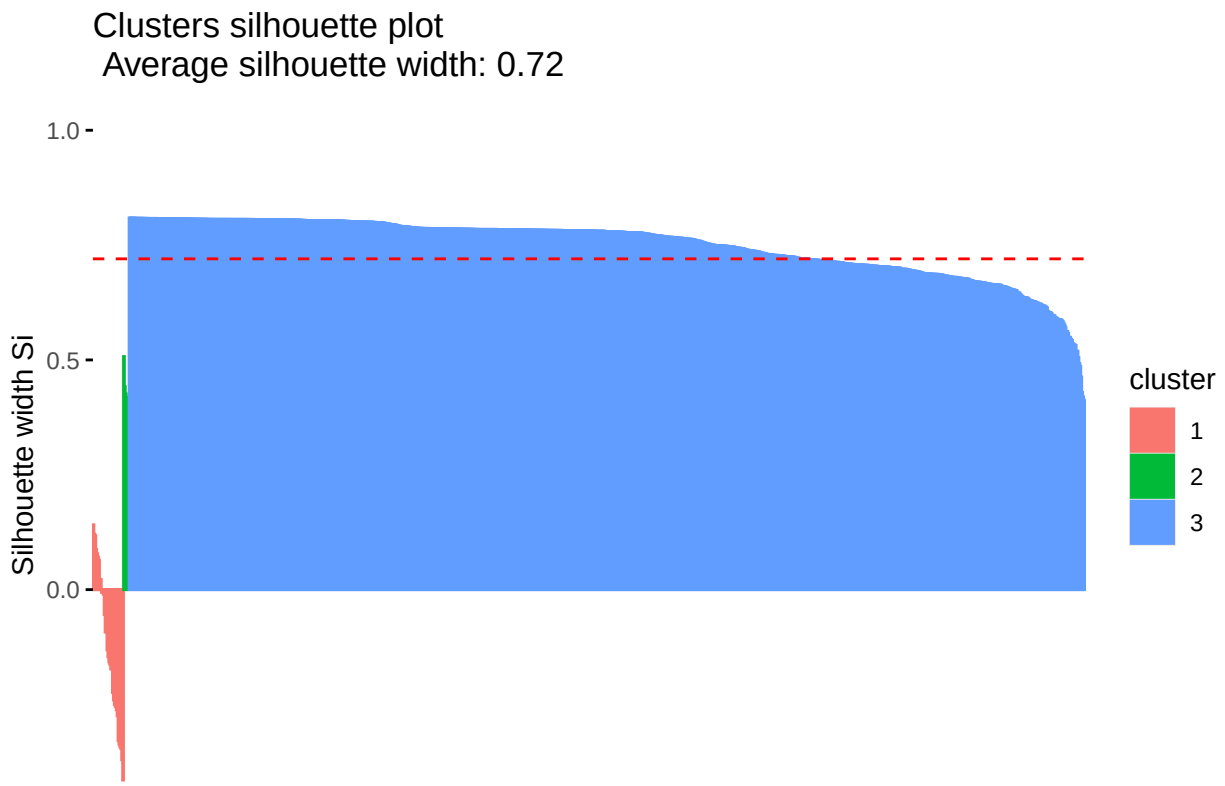



```
table(Kmeans = movies_clean$cluster_kmeans,
      Jerarquico = movies_clean$cluster_hc)
```

```
##       Jerarquico
## Kmeans    1    2    3
##      1    3   30    0
##      2    0    0    6
##      3 1033    3    0
```

```
sil_kmeans <- silhouette(kmeans_model$cluster, dist_matrix)
fviz_silhouette(sil_kmeans)
```

```
##   cluster size ave.sil.width
## 1       1   33         -0.13
## 2       2    6          0.40
## 3       3 1036          0.75
```

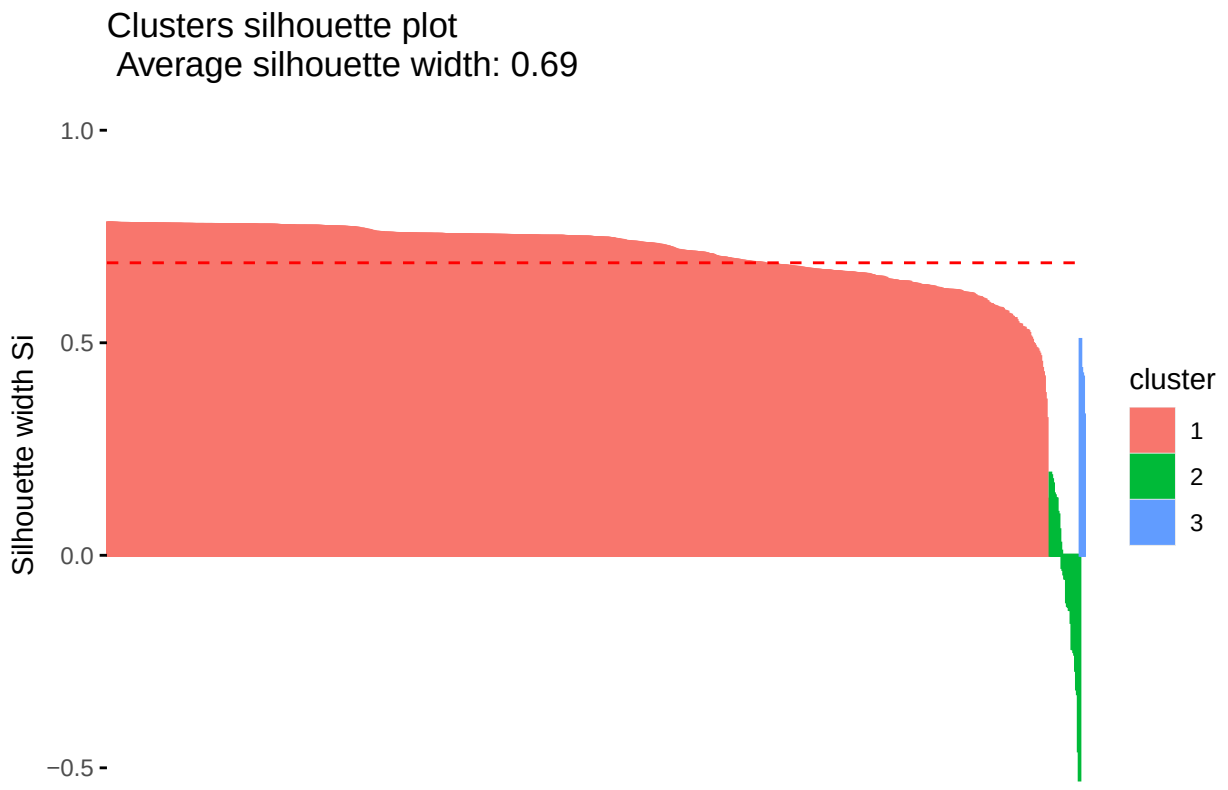



```
mean(sil_kmeans[,3])
```

```
## [1] 0.7200377
```

```
sil_hc <- silhouette(hc_clusters, dist_matrix)
fviz_silhouette(sil_hc)
```

```
##   cluster size ave.sil.width
## 1       1 1036          0.71
## 2       2   33         -0.07
## 3       3    6          0.40
```




```
mean(sil_hc[,3])
```

```
## [1] 0.688182
```

```
movies_clean %>%
  group_by(cluster_kmeans) %>%
  summarise(across(where(is.numeric),
                   list(media = mean,
                        mediana = median),
                   na.rm = TRUE))
```

```
## Warning: There was 1 warning in `summarise()`.
## i In argument: `across(...)`.
## i In group 1: `cluster_kmeans = 1`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))

## # A tibble: 3 x 29
```

```
##   cluster_kmeans popularity_media popularity_mediana budget_media budget_mediana
##   <fct>                     <dbl>              <dbl>        <dbl>          <dbl>
## 1 1                        21.4                17.0        34685.              0
## 2 2                        37.2                23.9          100               0
## 3 3                         0.184               0.04         194.              0
## # i 24 more variables: revenue_media <dbl>, revenue_mediana <dbl>,
## #   runtime_media <dbl>, runtime_mediana <dbl>, genresAmount_media <dbl>,
## #   genresAmount_mediana <dbl>, productionCoAmount_media <dbl>,
## #   productionCoAmount_mediana <dbl>, productionCountriesAmount_media <dbl>,
## #   productionCountriesAmount_mediana <dbl>, voteCount_media <dbl>,
## #   voteCount_mediana <dbl>, voteAvg_media <dbl>, voteAvg_mediana <dbl>,
## #   actorsPopularity_media <dbl>, actorsPopularity_mediana <dbl>, ...
```

```r
table(movies_clean$cluster_kmeans)
```

```
##
##    1    2    3
##   33    6 1036
```

2. Reglas de Asociación

2.1 Generación de reglas con algoritmo Apriori

```r
library(arules)
```

```
## Cargando paquete requerido: Matrix

##
## Adjuntando el paquete: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

##
## Adjuntando el paquete: 'arules'

## The following object is masked from 'package:dplyr':
##
##     recode

## The following objects are masked from 'package:base':
##
##     abbreviate, write
```

```r
library(arulesViz)

# Seleccionar variables numéricas
movies_rules <- movies_clean %>%
  select(popularity, budget, revenue, runtime,
         voteCount, voteAvg, actorsPopularity,
         actorsAmount, castWomenAmount,
         castMenAmount, releaseYear)

# Función segura de discretización en 3 niveles
discretize_safe <- function(x){
  breaks <- quantile(x, probs = seq(0, 1, length.out = 4), na.rm = TRUE)
  breaks <- unique(breaks)  # evitar duplicados
  cut(x,
      breaks = breaks,
      include.lowest = TRUE,
      labels = c("Low","Medium","High")[1:(length(breaks)-1)])
}

# Aplicar discretización
movies_rules_disc <- as.data.frame(
  lapply(movies_rules, discretize_safe)
)

# Convertir a transacciones
movies_trans <- as(movies_rules_disc, "transactions")

summary(movies_trans)
```

```
## transactions as itemMatrix in sparse format with
##  1075 rows (elements/itemsets/transactions) and
##  18 columns (items) and a density of 0.6111111
##
## most frequent items:
##       budget=Low      revenue=Low    voteCount=Low      voteAvg=Low
##             1075             1075             1075             1075
## actorsAmount=Low          (Other)
##             1075             6450
##
## element (itemset/transaction) length distribution:
## sizes
##   11
## 1075
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      11      11      11      11      11      11
##
## includes extended item information - examples:
##              labels  variables levels
## 1    popularity=Low popularity    Low
## 2 popularity=Medium popularity Medium
## 3   popularity=High popularity   High
##
```

```
## includes extended transaction information - examples:
##   transactionID
## 1             1
## 2             2
## 3             3
```

Reglas con soporte = 0.20 y confianza = 0.60

```
rules1 <- apriori(movies_trans,
                  parameter = list(supp = 0.20,
                                   conf = 0.60,
                                   minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5     0.2      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 215
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[18 item(s), 1075 transaction(s)] done [0.00s].
## sorting and recoding items ... [18 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 done [0.00s].
## writing ... [10104 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules1)
```

```
## set of 10104 rules
##
## rule length distribution (lhs + rhs):sizes
##    2    3    4    5    6    7    8    9
##  134  700 1834 2800 2618 1484  470   64
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   4.000   5.000   5.358   6.000   9.000
##
## summary of quality measures:
##     support         confidence        coverage           lift
##  Min.   :0.2270   Min.   :0.6844   Min.   :0.2270   Min.   :0.9208
```

```
##  1st Qu.:0.2540   1st Qu.:1.0000   1st Qu.:0.2567   1st Qu.:1.0000
##  Median :0.3005   Median :1.0000   Median :0.3181   Median :1.0000
##  Mean   :0.3469   Mean   :0.9709   Mean   :0.3590   Mean   :1.0002
##  3rd Qu.:0.3330   3rd Qu.:1.0000   3rd Qu.:0.3405   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.1396
##      count
##  Min.   : 244.0
##  1st Qu.: 273.0
##  Median : 323.0
##  Mean   : 372.9
##  3rd Qu.: 358.0
##  Max.   :1075.0
##
## mining info:
##          data ntransactions support confidence
##  movies_trans          1075     0.2        0.6
##                                                                                call
##  apriori(data = movies_trans, parameter = list(supp = 0.2, conf = 0.6, minlen = 2))
```

```
inspect(head(sort(rules1, by = "lift"), 10))
```

```
##      lhs                        rhs                 support confidence  coverage     lift count
## [1]  {popularity=Low}       => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [2]  {popularity=Low,
##       castMenAmount=Low}    => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [3]  {popularity=Low,
##       castWomenAmount=Low}  => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [4]  {popularity=Low,
##       actorsAmount=Low}     => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [5]  {popularity=Low,
##       voteAvg=Low}          => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [6]  {popularity=Low,
##       voteCount=Low}        => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [7]  {popularity=Low,
##       revenue=Low}          => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [8]  {popularity=Low,
##       budget=Low}           => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [9]  {popularity=Low,
##       castWomenAmount=Low,
##       castMenAmount=Low}    => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [10] {popularity=Low,
##       actorsAmount=Low,
##       castMenAmount=Low}    => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
```

Reglas con soporte = 0.10 y confianza = 0.70

```
rules2 <- apriori(movies_trans,
                  parameter = list(supp = 0.10,
                                   conf = 0.70,
                                   minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 107
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[18 item(s), 1075 transaction(s)] done [0.00s].
## sorting and recoding items ... [18 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 7 8 9 10

## Warning in apriori(movies_trans, parameter = list(supp = 0.1, conf = 0.7, :
## Mining stopped (maxlen reached). Only patterns up to a length of 10 returned!

##  done [0.00s].
## writing ... [21496 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules2)
```

```
## set of 21496 rules
##
## rule length distribution (lhs + rhs):sizes
##    2    3    4    5    6    7    8    9   10
##  133  804 2548 4844 5838 4508 2164  588   69
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   5.000   6.000   5.907   7.000  10.000
##
## summary of quality measures:
##     support         confidence        coverage           lift
##  Min.   :0.1042   Min.   :0.7193   Min.   :0.1042   Min.   :0.9678
##  1st Qu.:0.1265   1st Qu.:1.0000   1st Qu.:0.1265   1st Qu.:1.0000
##  Median :0.1851   Median :1.0000   Median :0.1851   Median :1.0000
##  Mean   :0.2355   Mean   :0.9809   Mean   :0.2416   Mean   :1.0031
##  3rd Qu.:0.2884   3rd Qu.:1.0000   3rd Qu.:0.3005   3rd Qu.:1.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.1396
##      count
##  Min.   : 112.0
##  1st Qu.: 136.0
##  Median : 199.0
##  Mean   : 253.2
##  3rd Qu.: 310.0
##  Max.   :1075.0
```

```
##
## mining info:
##          data ntransactions support confidence
##  movies_trans          1075     0.1        0.7
##                                                                                call
##  apriori(data = movies_trans, parameter = list(supp = 0.1, conf = 0.7, minlen = 2))
```

```
inspect(head(sort(rules2, by = "lift"), 10))
```

```
##      lhs                          rhs                 support confidence  coverage     lift count
## [1]  {popularity=Low}          => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [2]  {popularity=Low,
##       castMenAmount=Low}       => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [3]  {popularity=Low,
##       castWomenAmount=Low}     => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [4]  {popularity=Low,
##       actorsAmount=Low}        => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [5]  {popularity=Low,
##       voteAvg=Low}             => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [6]  {popularity=Low,
##       voteCount=Low}           => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [7]  {popularity=Low,
##       revenue=Low}             => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [8]  {popularity=Low,
##       budget=Low}              => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [9]  {popularity=Low,
##       castWomenAmount=Low,
##       castMenAmount=Low}       => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
## [10] {popularity=Low,
##       actorsAmount=Low,
##       castMenAmount=Low}       => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.139573   310
```

Eliminación de ítems muy frecuentes (>80%)

```
item_freq <- itemFrequency(movies_trans)

freq_items <- names(item_freq[item_freq > 0.8])

movies_trans_reduced <- movies_trans[, !(colnames(movies_trans) %in% freq_items)]

rules3 <- apriori(movies_trans_reduced,
                  parameter = list(supp = 0.10,
                                   conf = 0.70,
                                   minlen = 2))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
```

```
##         0.7    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen target  ext
##      10  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 107
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[11 item(s), 1075 transaction(s)] done [0.00s].
## sorting and recoding items ... [11 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [13 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
summary(rules3)
```

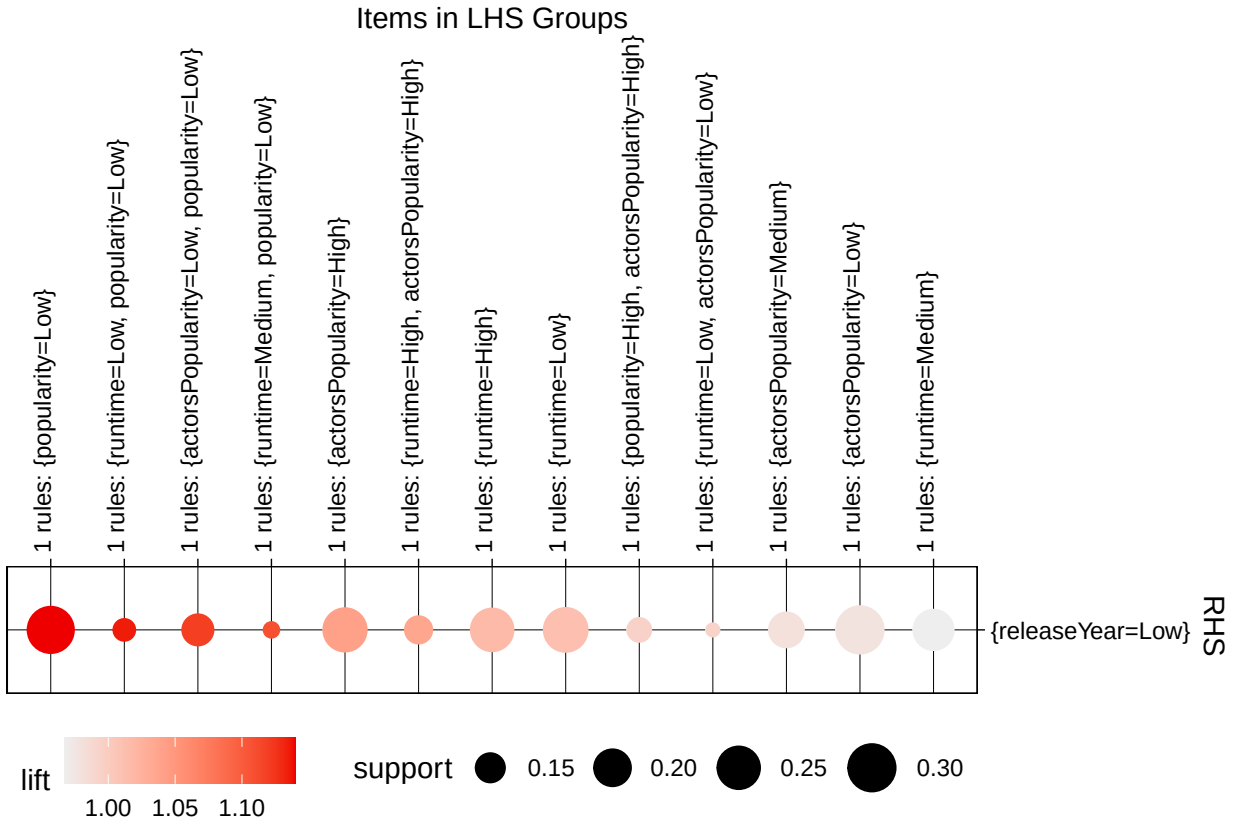
```
## set of 13 rules
##
## rule length distribution (lhs + rhs):sizes
## 2 3
## 7 6
##
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.000   2.000   2.000   2.462   3.000   3.000
##
## summary of quality measures:
##     support         confidence        coverage           lift
##  Min.   :0.1079   Min.   :0.7193   Min.   :0.1321   Min.   :0.9678
##  1st Qu.:0.1265   1st Qu.:0.7389   1st Qu.:0.1712   1st Qu.:0.9941
##  Median :0.1851   Median :0.7563   Median :0.2540   Median :1.0176
##  Mean   :0.1953   Mean   :0.7733   Mean   :0.2545   Mean   :1.0404
##  3rd Qu.:0.2577   3rd Qu.:0.8239   3rd Qu.:0.3330   3rd Qu.:1.1086
##  Max.   :0.3005   Max.   :0.8470   Max.   :0.4130   Max.   :1.1396
##      count
##  Min.   :116
##  1st Qu.:136
##  Median :199
##  Mean   :210
##  3rd Qu.:277
##  Max.   :323
##
## mining info:
##                  data ntransactions support confidence
##  movies_trans_reduced          1075     0.1        0.7
##                                                                                call
##  apriori(data = movies_trans_reduced, parameter = list(supp = 0.1, conf = 0.7, minlen = 2))
```

```
inspect(head(sort(rules3, by = "lift"), 10))
```

```
##      lhs                           rhs                 support confidence  coverage      lift count
## [1]  {popularity=Low}           => {releaseYear=Low} 0.2883721  0.8469945 0.3404651 1.1395734   310
## [2]  {popularity=Low,
##       runtime=Low}              => {releaseYear=Low} 0.1200000  0.8431373 0.1423256 1.1343837   129
## [3]  {popularity=Low,
##       actorsPopularity=Low}     => {releaseYear=Low} 0.1609302  0.8317308 0.1934884 1.1190370   173
## [4]  {popularity=Low,
##       runtime=Medium}           => {releaseYear=Low} 0.1088372  0.8239437 0.1320930 1.1085600   117
## [5]  {actorsPopularity=High}   => {releaseYear=Low} 0.2576744  0.7737430 0.3330233 1.0410185   277
## [6]  {runtime=High,
##       actorsPopularity=High}   => {releaseYear=Low} 0.1404651  0.7704082 0.1823256 1.0365316   151
## [7]  {runtime=High}            => {releaseYear=Low} 0.2511628  0.7563025 0.3320930 1.0175535   270
## [8]  {runtime=Low}             => {releaseYear=Low} 0.2632558  0.7526596 0.3497674 1.0126521   283
## [9]  {popularity=High,
##       actorsPopularity=High}   => {releaseYear=Low} 0.1265116  0.7391304 0.1711628 0.9944496   136
## [10] {runtime=Low,
##       actorsPopularity=Low}    => {releaseYear=Low} 0.1079070  0.7388535 0.1460465 0.9940770   116
```

Visualización

```
plot(rules3, method = "grouped")
```



Discusión

Con soporte 0.20 y confianza 0.60 se obtienen reglas generales del comportamiento del dataset.

Al reducir el soporte a 0.10 y aumentar la confianza a 0.70 se obtienen reglas más específicas y fuertes.

Eliminar ítems muy frecuentes mejora la calidad de las reglas y evita asociaciones triviales.

Las reglas con mayor lift representan asociaciones no triviales entre variables como presupuesto, ingresos y popularidad.

Analisis de Componentes Principales PCA

¿Se pueden incluir variables categoricas?

Las variables categóricas como:

-originalLanguage

-genres

-productionCountry

-director

-productionCompany

No pueden incluirse directamente en PCA porque:

-PCA trabaja con matriz de covarianza/correlación

-Requiere variables numéricas continuas

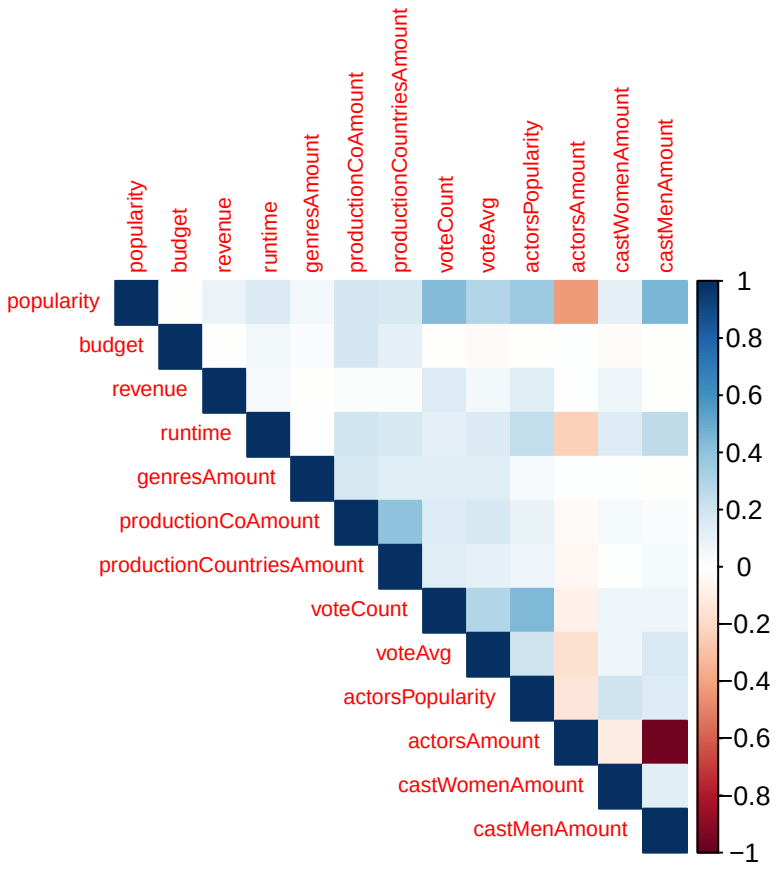
-Transformarlas con One-Hot Encoding generaría cientos de variables

-Alta cardinalidad → distorsiona la varianza

¿Es conveniente aplicar PCA?

Trabajando solo con variables numericas:

```
##
## Adjuntando el paquete: 'psych'

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

## corrplot 0.95 loaded

## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(everything(), as.numeric)`.
## Caused by warning:
## ! NAs introducidos por coerción
```




```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = cor_matrix)
## Overall MSA =  0.66
## MSA for each item =
##                 popularity                     budget                    revenue
##                       0.80                       0.63                       0.79
##                    runtime                genresAmount         productionCoAmount
##                       0.77                       0.69                       0.66
## productionCountriesAmount                  voteCount                    voteAvg
##                       0.67                       0.70                       0.84
##           actorsPopularity                actorsAmount           castWomenAmount
##                       0.74                       0.57                       0.75
##              castMenAmount
##                       0.56

## $chisq
## [1] 4630.2
##
## $p.value
## [1] 0
##
## $df
## [1] 78
```

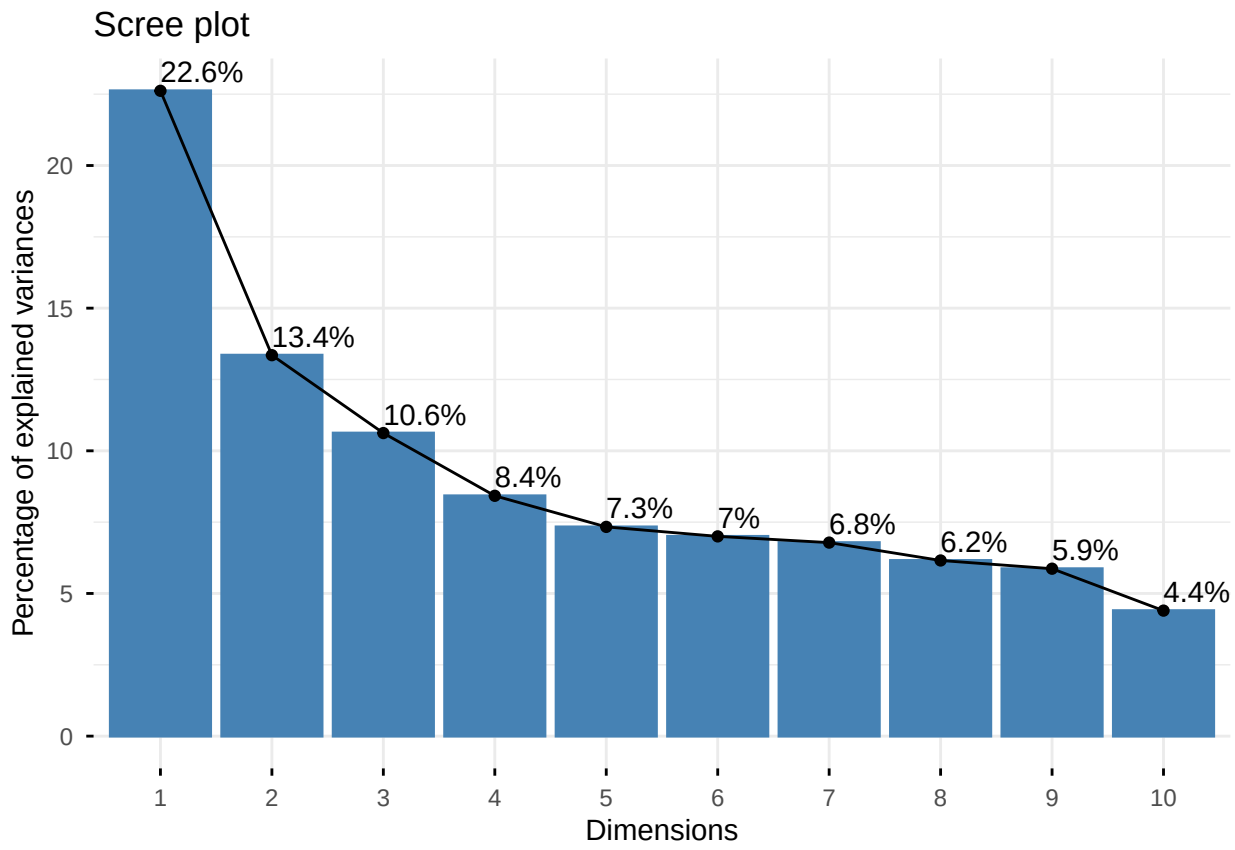
La matriz de correlación evidencia dependencias lineales moderadas y fuertes entre varias variables, lo que justifica la aplicación de PCA para reducir dimensionalidad. El índice KMO global de 0.66 indica que la

estructura de correlaciones es aceptable para aplicar análisis factorial o PCA. El test de esfericidad de Bartlett resultó altamente significativo ($p < 0.001$), lo que confirma que la matriz de correlación no es identidad y que existen correlaciones suficientes para aplicar PCA. Se seleccionaron los primeros 4 componentes principales, ya que presentan valores propios mayores a 1 y explican aproximadamente el 55% de la variabilidad total.

Aplicacion del PCA

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     1.7148 1.3175 1.1751 1.04644 0.97640 0.95403 0.93904
## Proportion of Variance 0.2262 0.1335 0.1062 0.08423 0.07334 0.07001 0.06783
## Cumulative Proportion  0.2262 0.3597 0.4659 0.55017 0.62350 0.69352 0.76135
##                           PC8     PC9    PC10    PC11   PC12    PC13
## Standard deviation     0.89463 0.87342 0.75619 0.70635 0.6629 0.17050
## Proportion of Variance 0.06157 0.05868 0.04399 0.03838 0.0338 0.00224
## Cumulative Proportion  0.82291 0.88160 0.92558 0.96396 0.9978 1.00000
```

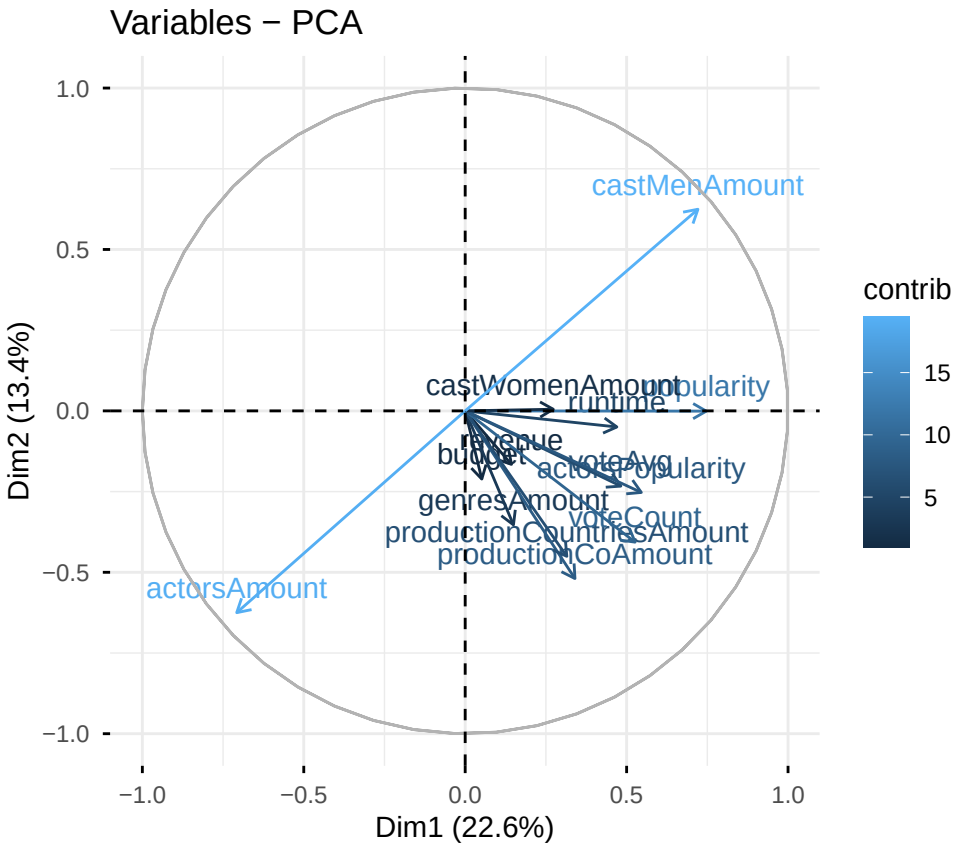
```
## Warning in geom_bar(stat = "identity", fill = barfill, color = barcolor, :
## Ignoring empty aesthetic: `width`.
```

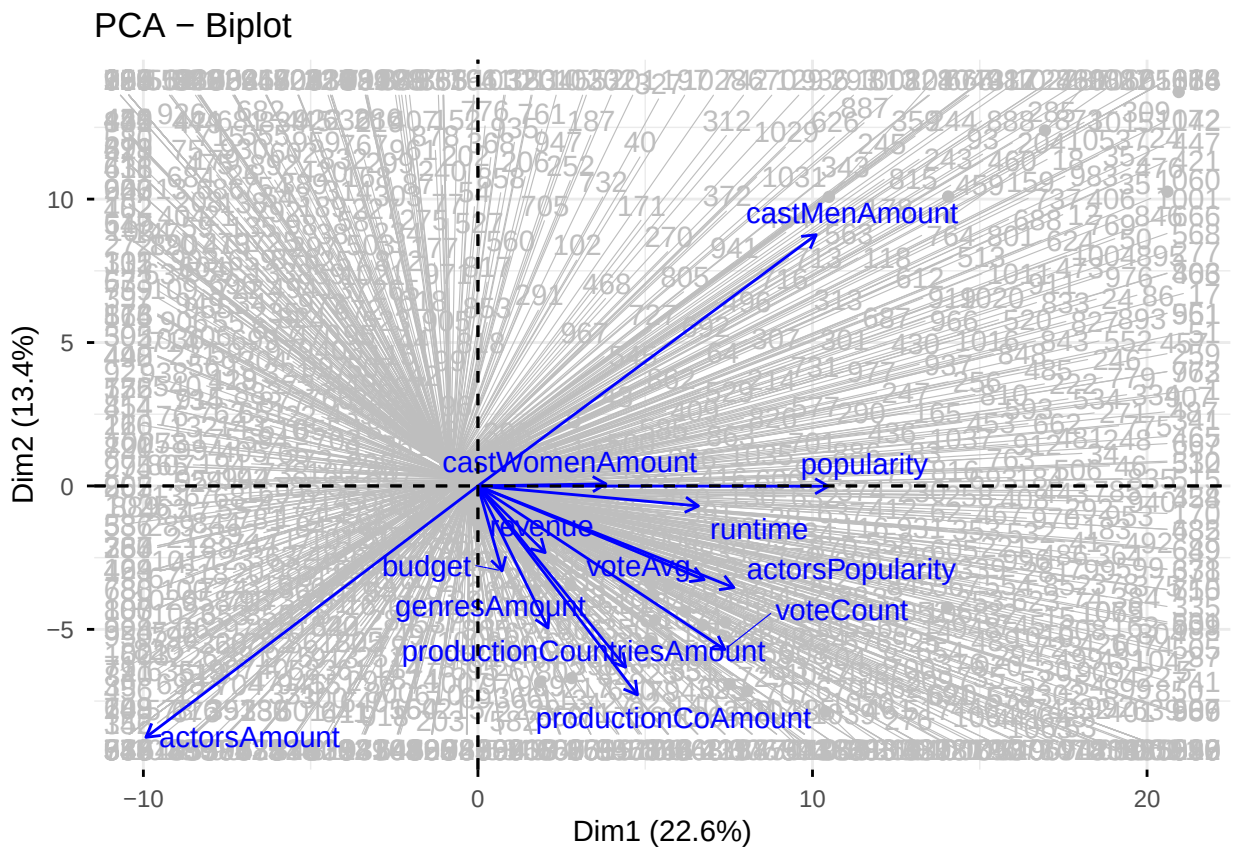



```
##                                PC1           PC2        PC3         PC4
## popularity            0.43542868 -0.0005943905  0.1126876  0.17617442
## budget                0.02997437 -0.1602710629 -0.3761440 -0.28149110
## revenue               0.08359436 -0.1257894030  0.2979818 -0.27104635
```

```
## runtime                    0.27399198 -0.0377905466 -0.1459826 -0.41236679
## genresAmount               0.08715012 -0.2679096337 -0.1316260  0.45333878
## productionCoAmount         0.19825057 -0.3941890154 -0.4130260 -0.06164206
## productionCountriesAmount  0.18355760 -0.3420069390 -0.4116726 -0.04580917
## voteCount                  0.30696326 -0.3085546127  0.3603822  0.17345427
## voteAvg                    0.28113008 -0.1768463787  0.1086102  0.30071303
## actorsPopularity           0.31813996 -0.1917610680  0.3810032 -0.18535925
## actorsAmount              -0.41269398 -0.4743279116  0.1756529 -0.07105981
## castWomenAmount            0.15909004  0.0038582521  0.1772479 -0.52135099
## castMenAmount              0.42041931  0.4740017764 -0.1709301  0.05540265
##                                   PC5          PC6         PC7         PC8
## popularity                 0.157559615 -0.127312615 -0.01076350 -0.28624741
## budget                     0.480312399 -0.098090786 -0.67947694  0.14593230
## revenue                    0.526069546  0.665458564  0.27903130  0.12206156
## runtime                   -0.240979426 -0.145207262  0.23600335  0.59991198
## genresAmount              -0.291973025  0.546011173 -0.34514584  0.12045604
## productionCoAmount        -0.050127656  0.022183150  0.14772374 -0.11924923
## productionCountriesAmount  0.005961045  0.007171287  0.38237936 -0.35839839
## voteCount                  0.143992389 -0.197268630 -0.09558008 -0.11552709
## voteAvg                   -0.111308387 -0.024802732  0.01813402  0.50480040
## actorsPopularity          -0.025557041 -0.254596533 -0.13662058 -0.02947734
## actorsAmount              -0.068741713 -0.113075913  0.01982208  0.02852342
## castWomenAmount           -0.529146321  0.287800197 -0.29746286 -0.30682904
## castMenAmount              0.057968715  0.107764224 -0.02169366 -0.02822801
##                                   PC9          PC10        PC11         PC12
## popularity                 0.04474196 -0.0976154616  0.21432457  0.764664825
## budget                     0.08016029  0.1430284962  0.03374290  0.036938109
## revenue                    0.02148098 -0.0300767463 -0.02465417  0.024738445
## runtime                   -0.35090728  0.0100551672  0.30210473  0.150024941
## genresAmount              -0.41315762  0.0335523672 -0.02401737  0.091462226
## productionCoAmount         0.18498095 -0.7251319870 -0.11276150 -0.138164226
## productionCountriesAmount -0.03443635  0.6259899218 -0.08154153 -0.057341237
## voteCount                 -0.12167118 -0.0003315423  0.53472861 -0.517081809
## voteAvg                    0.67180256  0.2098499612 -0.15445571 -0.006339279
## actorsPopularity          -0.31176774 -0.0031653165 -0.70670043 -0.029053038
## actorsAmount               0.02293111  0.0054241540  0.07151350  0.244788720
## castWomenAmount            0.31124569  0.0776611299  0.15141348 -0.009840161
## castMenAmount             -0.03392206 -0.0097469870 -0.06207499 -0.178283172
##                                   PC13
## popularity                -0.037425563
## budget                    -0.000603145
## revenue                    0.001447045
## runtime                   -0.019570362
## genresAmount              -0.002987639
## productionCoAmount         0.005923534
## productionCountriesAmount  0.005332753
## voteCount                  0.019048033
## voteAvg                    0.007568777
## actorsPopularity          -0.002785658
## actorsAmount               0.696049007
## castWomenAmount           -0.008413845
## castMenAmount              0.716350844

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
```

```
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the ggpubr package.
##   Please report the issue at <https://github.com/kassambara/ggpubr/issues>.
## This warning is displayed once per session.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```





El análisis de componentes principales permitió identificar cuatro dimensiones fundamentales en el desempeño de las películas: impacto comercial, estructura del elenco, complejidad productiva e inversión económica.

Esta reducción facilita la construcción futura de modelos predictivos, disminuye problemas de colinealidad y permite segmentar películas en función de características estructurales más compactas.

4. Otros Algoritmos de Aprendizaje No Supervisado

4.1 Selección del algoritmo

Para este apartado se decidió utilizar **UMAP (Uniform Manifold Approximation and Projection)**.

UMAP es una técnica de reducción de dimensionalidad no lineal que permite proyectar datos de alta dimensión en un espacio de menor dimensión preservando tanto la estructura local como parte de la estructura global.

Se eligió este algoritmo debido a que:

- El dataset contiene múltiples variables numéricas relevantes (budget, revenue, popularity, voteAvg, voteCount, runtime, actorsPopularity, etc.).
- El conjunto de datos es grande (~19,883 observaciones), y UMAP es más eficiente computacionalmente que t-SNE.
- Permite detectar estructuras complejas o agrupamientos no lineales que PCA no logra capturar.
- Facilita la visualización de posibles segmentos naturales de películas.

4.2 Preparación de los datos

Se utilizaron únicamente variables numéricas relevantes para evitar ruido proveniente de identificadores o variables textuales.

```
library(dplyr)
library(umap)
library(ggplot2)
library(scales)
```

```
##
## Adjuntando el paquete: 'scales'

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

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

```
movies_num <- movies %>%
  select(popularity, budget, revenue, runtime,
         voteAvg, voteCount, actorsPopularity,
         actorsAmount, castWomenAmount, castMenAmount,
         productionCoAmount, productionCountriesAmount) %>%
  # Aseguramos que TODO sea numérico antes de omitir NA
  mutate(across(everything(), ~as.numeric(as.character(.)))) %>%
  na.omit()
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across(everything(), ~as.numeric(as.character(.)))`.
## Caused by warning:
## ! NAs introducidos por coerción
```

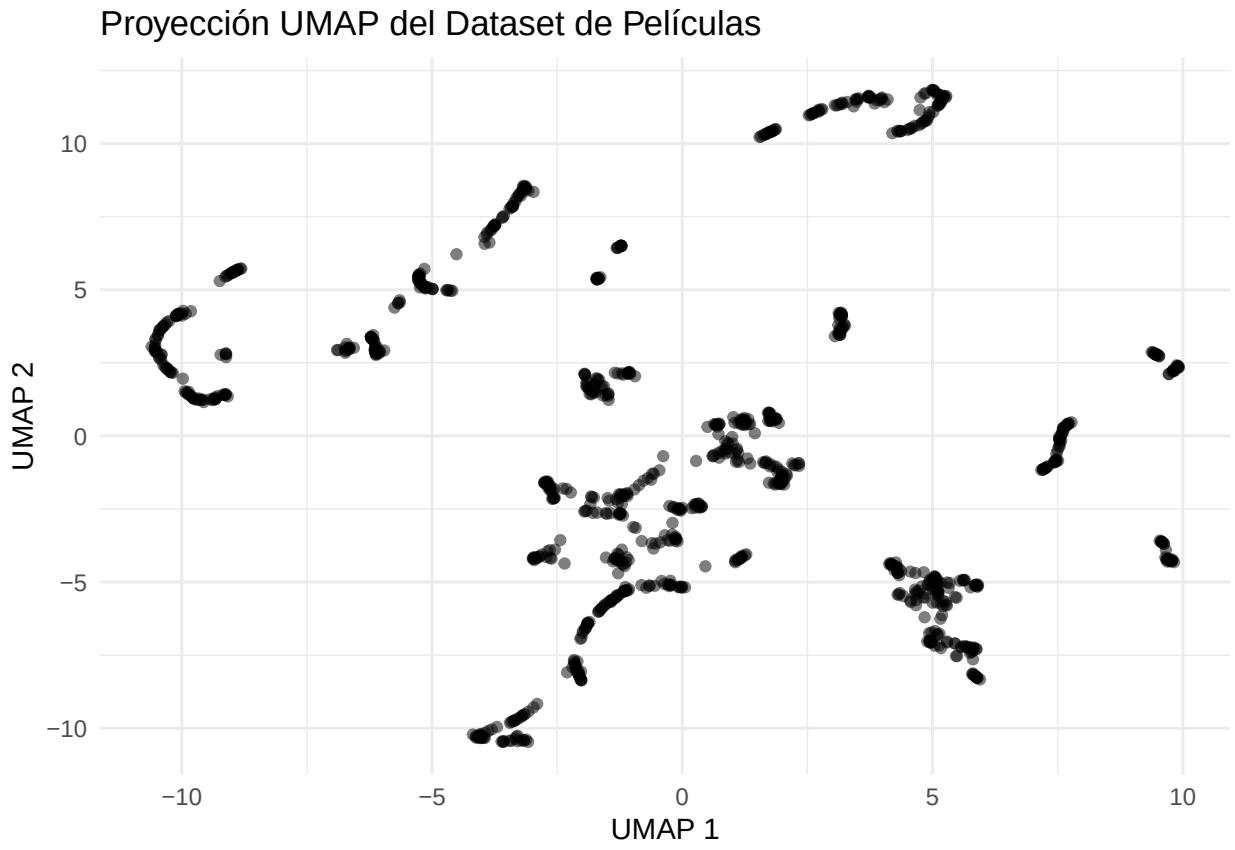
El escalamiento es necesario debido a que variables como *revenue* y *budget* tienen magnitudes muy superiores a otras variables como *voteAvg*, lo que podría sesgar la proyección.

4.3 Aplicación de UMAP

```
set.seed(123)

umap_result <- umap(movies_scaled)

umap_df <- as.data.frame(umap_result$layout)
colnames(umap_df) <- c("UMAP1", "UMAP2")
```

4.4 Visualización de resultados

```
ggplot(umap_df, aes(x = UMAP1, y = UMAP2)) +
  geom_point(alpha = 0.5) +
  theme_minimal() +
  labs(title = "Proyección UMAP del Dataset de Películas",
       x = "UMAP 1",
       y = "UMAP 2")
```



4.5 Interpretación de resultados

La proyección obtenida mediante UMAP permite observar:

- La existencia de regiones densas que sugieren segmentos naturales de películas.
- Separaciones claras entre grupos de películas con características financieras y de popularidad similares.
- Posibles conglomerados asociados a:
  - Películas de alto presupuesto y alto revenue.
  - Películas independientes de bajo presupuesto.

  - Películas con alto voteCount y alta popularidad del elenco.
  - Producciones con múltiples compañías y coproducciones internacionales.

A diferencia del PCA, UMAP captura estructuras no lineales, lo cual permite visualizar mejor relaciones complejas entre variables financieras, de popularidad y de producción.

4.6 Relevancia para CineVision Studios

El uso de UMAP aporta valor estratégico porque:

- Permite identificar segmentos de mercado no evidentes.
- Facilita detectar nichos como películas altamente rentables con bajo presupuesto.
- Permite visualizar patrones relacionados con popularidad del elenco y éxito financiero.
- Complementa los resultados obtenidos en el clustering, validando visualmente la existencia de agrupamientos naturales.

En conclusión, UMAP resulta ser una herramienta relevante para este conjunto de datos, ya que revela estructuras complejas que pueden ser explotadas por CineVision Studios para la toma de decisiones estratégicas en futuras producciones.