Proyecto1

Juan Luis Solórzano (carnet: 201598) Micaela Yataz (carnet: 18960)

2025-01-20

git: https://github.com/JusSolo/Mineria_Proyecto1.2.git

```
##        id             budget             genres            homePage
##  Min.   :     5   Min.   :        0   Length:10000       Length:10000
##  1st Qu.: 12286   1st Qu.:        0   Class :character   Class :character
##  Median :152558   Median :   500000   Mode  :character   Mode  :character
##  Mean   :249877   Mean   : 18551632
##  3rd Qu.:452022   3rd Qu.: 20000000
##  Max.   :922260   Max.   :380000000
##  productionCompany  productionCompanyCountry productionCountry
##  Length:10000       Length:10000             Length:10000
##  Class :character   Class :character         Class :character
##  Mode  :character   Mode  :character         Mode  :character
##
##
##
##     revenue             runtime        video           director
##  Min.   :0.000e+00   Min.   :  0.0   Mode :logical   Length:10000
##  1st Qu.:0.000e+00   1st Qu.: 90.0   FALSE:9430      Class :character
##  Median :1.631e+05   Median :100.0   TRUE :84        Mode  :character
##  Mean   :5.674e+07   Mean   :100.3   NA's :486
##  3rd Qu.:4.480e+07   3rd Qu.:113.0
##  Max.   :2.847e+09   Max.   :750.0
##     actors          actorsPopularity   actorsCharacter    originalTitle
##  Length:10000       Length:10000       Length:10000       Length:10000
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##     title           originalLanguage     popularity        releaseDate
##  Length:10000       Length:10000       Min.   :    4.258   Length:10000
##  Class :character   Class :character   1st Qu.:   14.578   Class :character
##  Mode  :character   Mode  :character   Median :   21.906   Mode  :character
##                                        Mean   :   51.394
##                                        3rd Qu.:   40.654
##                                        Max.   :11474.647
##     voteAvg         voteCount      genresAmount    productionCoAmount
##  Min.   : 1.300   Min.   :    1   Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 5.900   1st Qu.:  120   1st Qu.: 2.000   1st Qu.: 2.000
##  Median : 6.500   Median :  415   Median : 3.000   Median : 3.000
##  Mean   : 6.483   Mean   : 1342   Mean   : 2.596   Mean   : 3.171
```

```
##  3rd Qu.: 7.200   3rd Qu.: 1316   3rd Qu.: 3.000   3rd Qu.: 4.000
##  Max.   :10.000   Max.   :30788   Max.   :16.000   Max.   :89.000
##  productionCountriesAmount  actorsAmount    castWomenAmount
##  Min.   :  0.000           Min.   :     0   Length:10000
##  1st Qu.:  1.000           1st Qu.:    13   Class :character
##  Median :  1.000           Median :    21   Mode  :character
##  Mean   :  1.751           Mean   :  2148
##  3rd Qu.:  2.000           3rd Qu.:    36
##  Max.   :155.000           Max.   :919590
##  castMenAmount
##  Length:10000
##  Class :character
##  Mode  :character
##
##
##
```

1. Clustering

1.1. Haga el preprocesamiento del dataset, explique qué variables no aportan información a la generación de grupos y por qué. Describa con qué variables calculará los grupos.

Como el algoritmo de k-medias y el clustering jerárquico necesitan de alguna medida de distancia, entre los datos, en una primera instancia vamos a tomas solo las variables numéricas y vamos a quitar el id por ser como el nombre de una película. Las variables que tomaremos en consideración son las siguientes:

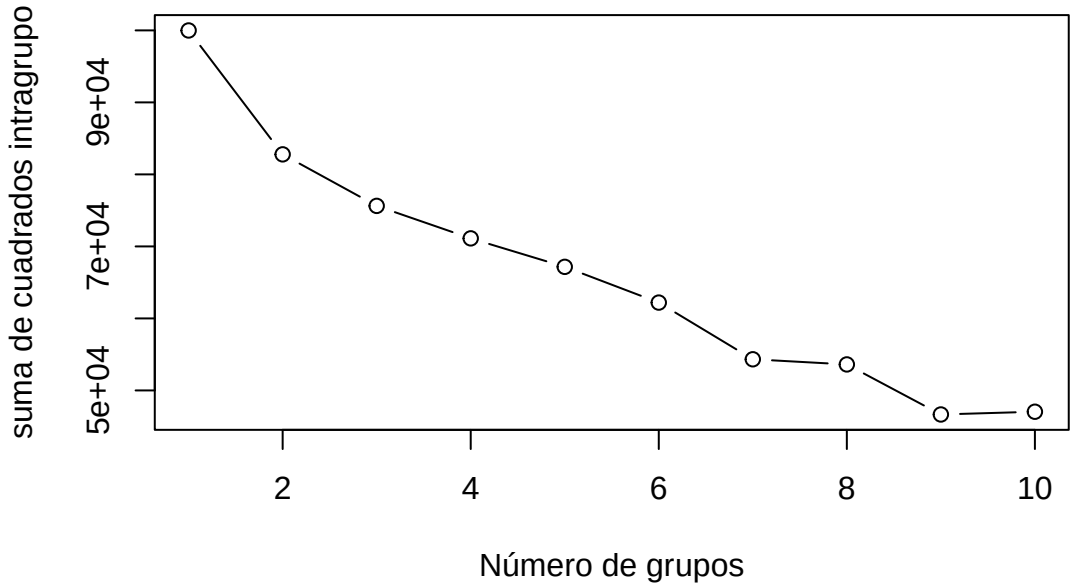
```
## 'data.frame':    10000 obs. of  10 variables:
##  $ budget                   : int  4000000 21000000 11000000 94000000 55000000 15000000 839727 12800
##  $ revenue                  : num  4.26e+06 1.21e+07 7.75e+08 9.40e+08 6.77e+08 ...
##  $ runtime                  : int  98 110 121 100 142 122 119 141 126 149 ...
##  $ popularity               : num  20.9 9.6 100 134.4 58.8 ...
##  $ voteAvg                  : num  5.7 6.5 8.2 7.8 8.5 8 8 7.9 7.5 8.2 ...
##  $ voteCount                : int  2077 223 16598 15928 22045 9951 4253 1335 8726 1963 ...
##  $ genresAmount             : int  2 3 3 2 3 1 2 2 5 2 ...
##  $ productionCoAmount       : int  2 3 2 1 2 2 2 26 2 1 ...
##  $ productionCountriesAmount: int  1 2 1 1 1 1 1 12 1 1 ...
##  $ actorsAmount             : int  25 15 105 24 76 40 152 29 117 24 ...
```

1.2. Analice la tendencia al agrupamiento usando el estadístico de Hopkings y la VAT (Visual Assessment of cluster Tendency). Esta última hágala si es posible, teniendo en cuenta las dimensiones del conjunto de datos. Discuta sus resultados e impresiones.

El estadístico de Hopkings es de 1 que es lejano a 0.5, entonces los datos no son aleatorios. Sin embargo no haremos un VAT por ser difícil de visualizar e interpretar con 10 variables.

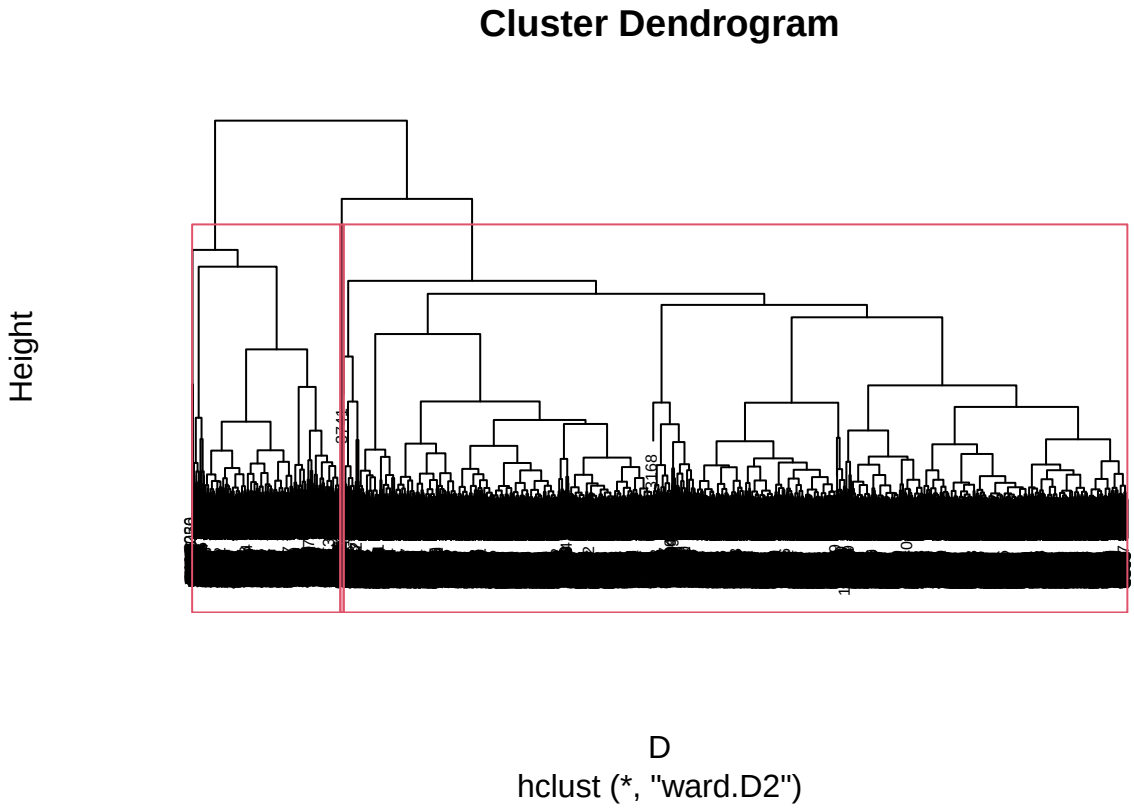
1.3. Determine cuál es el número de grupos a formar más adecuado para los datos que está trabajando. Haga una gráfica de codo y explique la razón de la elección de la cantidad de clústeres con la que trabajará.

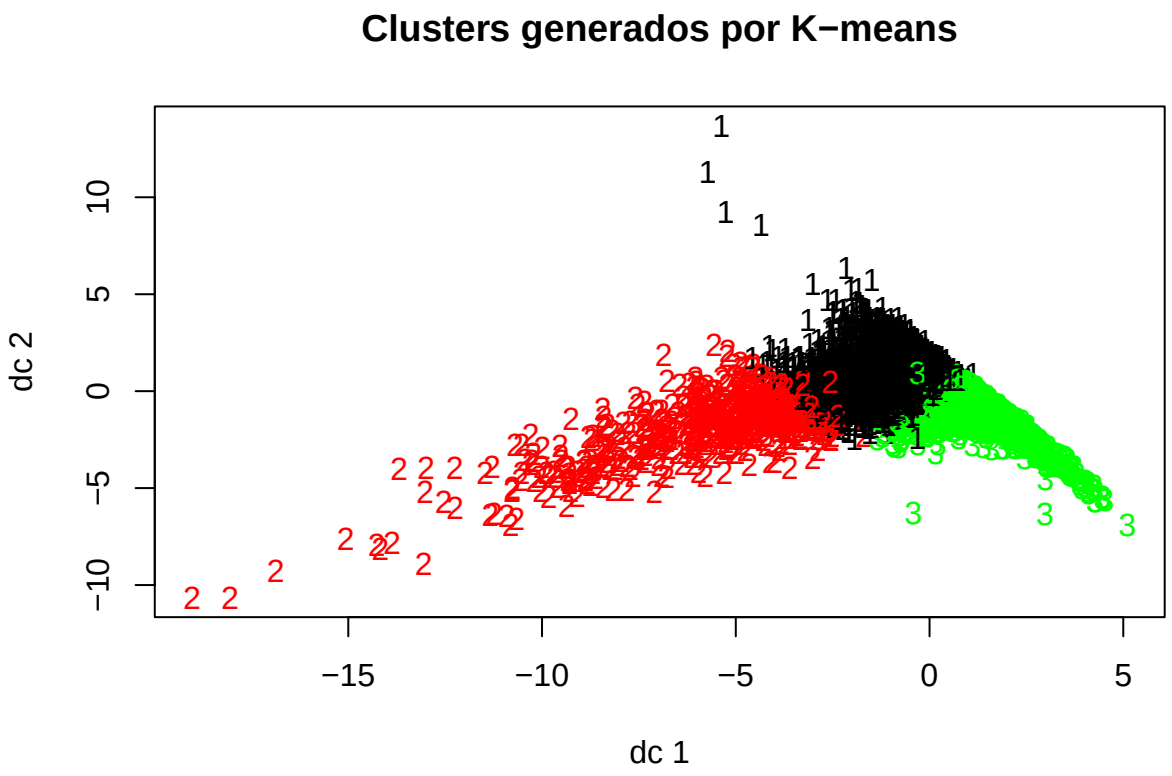
Para ello se usará el metodo de Codo

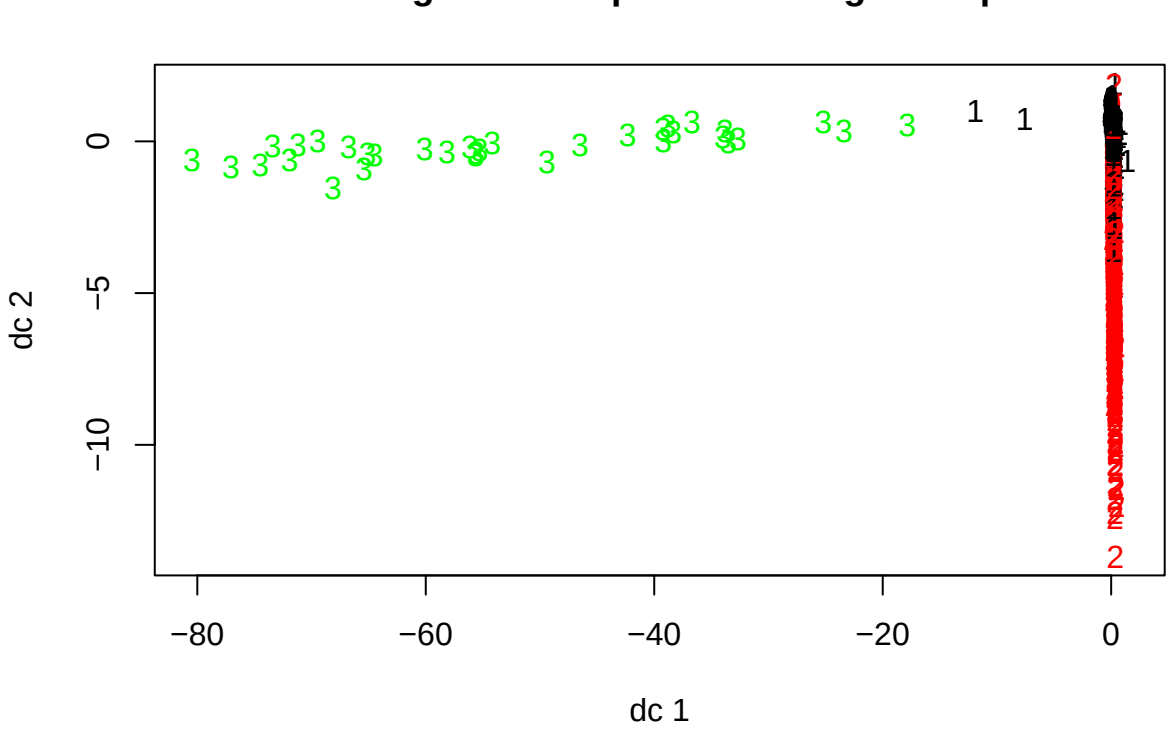


Como a partir de 3 grupos en adelante la suma de cuadrados intragrupo no disminuye tan rápido se elegirán 3 grupos.

1.4. Utilice los algoritmos k-medias y clustering jerárquico para agrupar. Compare los resultados generados por cada uno.

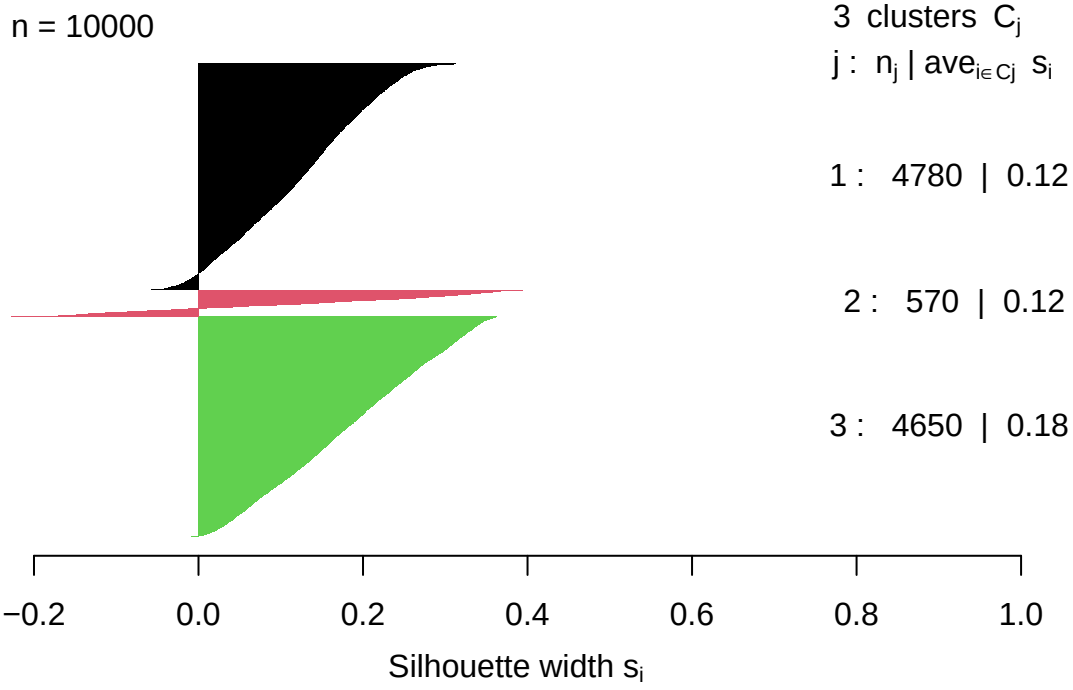



Clusters generados por K−means
dc 2
10
5
0
−5
−10
−15
−10
−5
0
5
dc 1


Clusters generados por Clustering Jerárquico
dc 2
0
−5
−10
−80
−60
−40
−20
0
dc 1

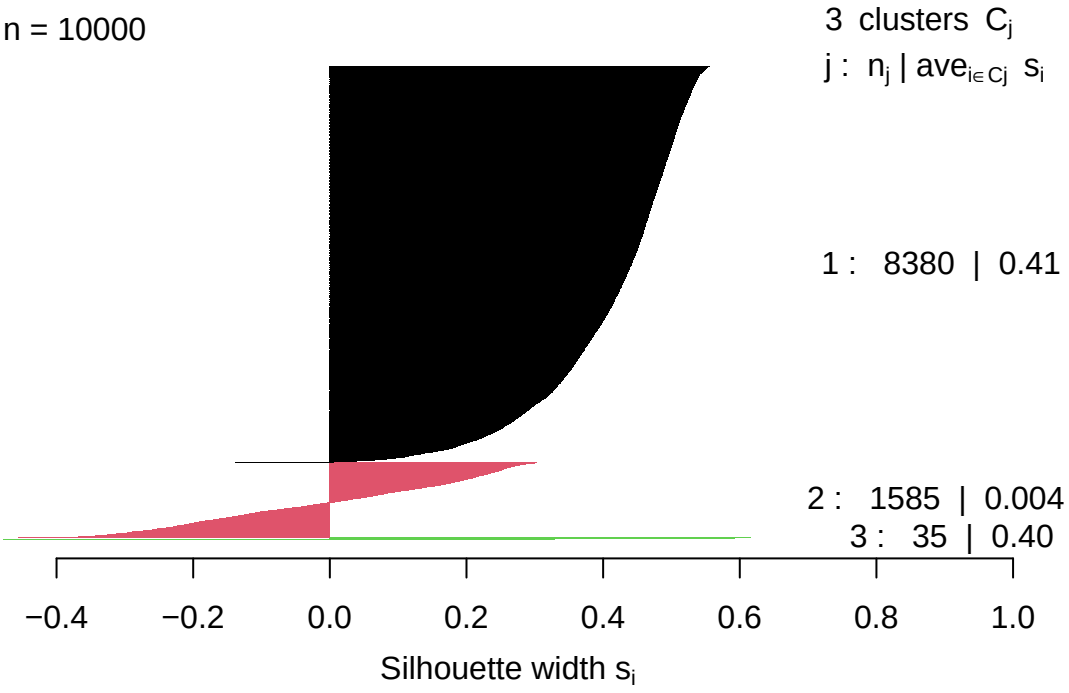
1.5. Determine la calidad del agrupamiento hecho por cada algoritmo con el método de la silueta. Discuta los resultados.

Silueta K−means



Average silhouette width : 0.15

Silueta Clustering Jerárquico



Average silhouette width : 0.35

Usando el método de la silueta el clustering jerárquico que tiene una silueta promedio de $0.35 > 0.15$ del clustering de Kmedias. Entonces para estos datos el clusterin jerárqioco obtubo un mejor resultado. Esto tiene sentido pues en la entrega anterior vimos que ninguna variable se comportaba de manera normal. En estos casos el k-medias no suele ser tan eficiente.

1.6. Interprete los grupos basado en el conocimiento que tiene de los datos. Recuerde investigar las medidas de tendencia central de las variables continuas y las tablas de frecuencia de las variables categóricas pertenecientes a cada grupo. Identifique hallazgos interesantes debido a las agrupaciones y describa para qué le podría servir.

```
## Tamaños de los dataframes:
```

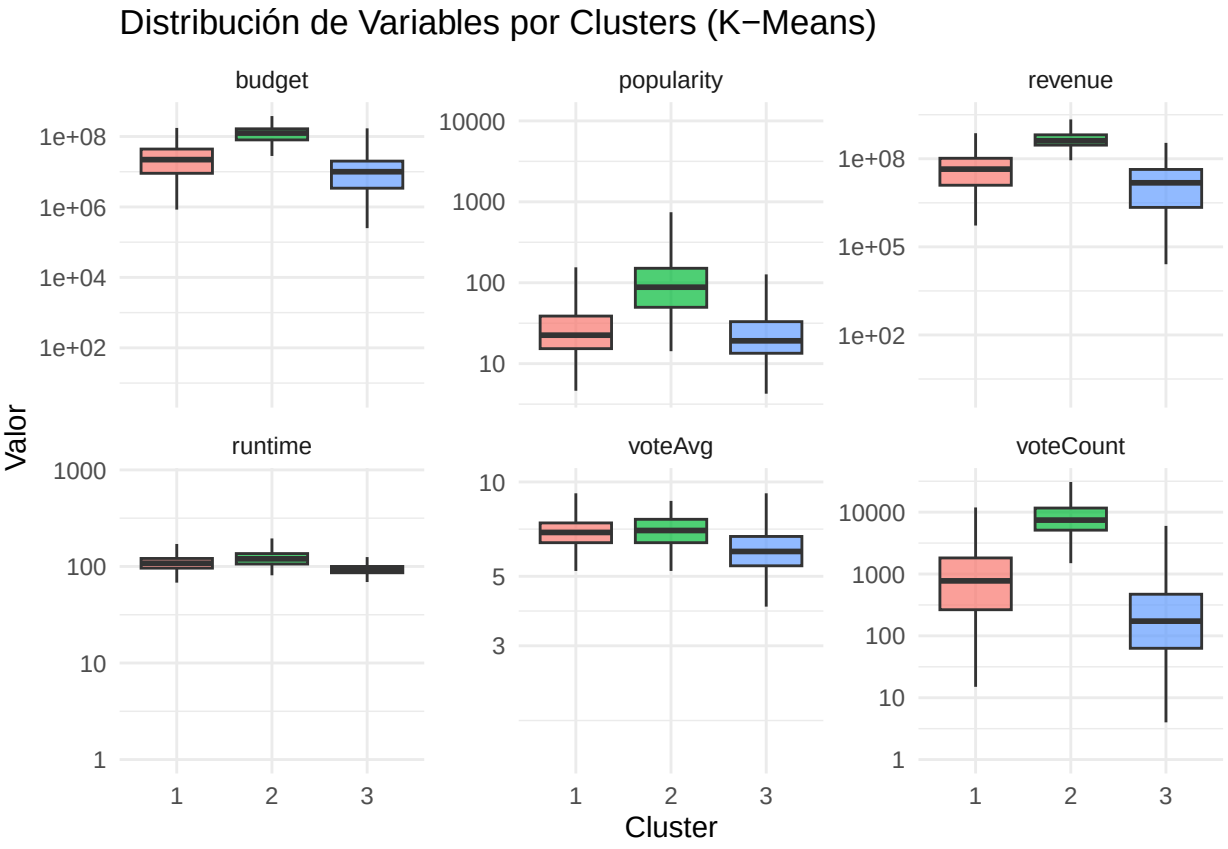
```
## K-Means -> Cluster 1: 4780  | Cluster 2: 570  | Cluster 3: 4650
```

```
## Jerárquico -> Cluster 1: 8380  | Cluster 2: 1585  | Cluster 3: 35
```

Como es dificil interpretar los numeros puros vamos a hacer unas gráficas de caja y bigote.

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## Warning: Removed 9688 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```

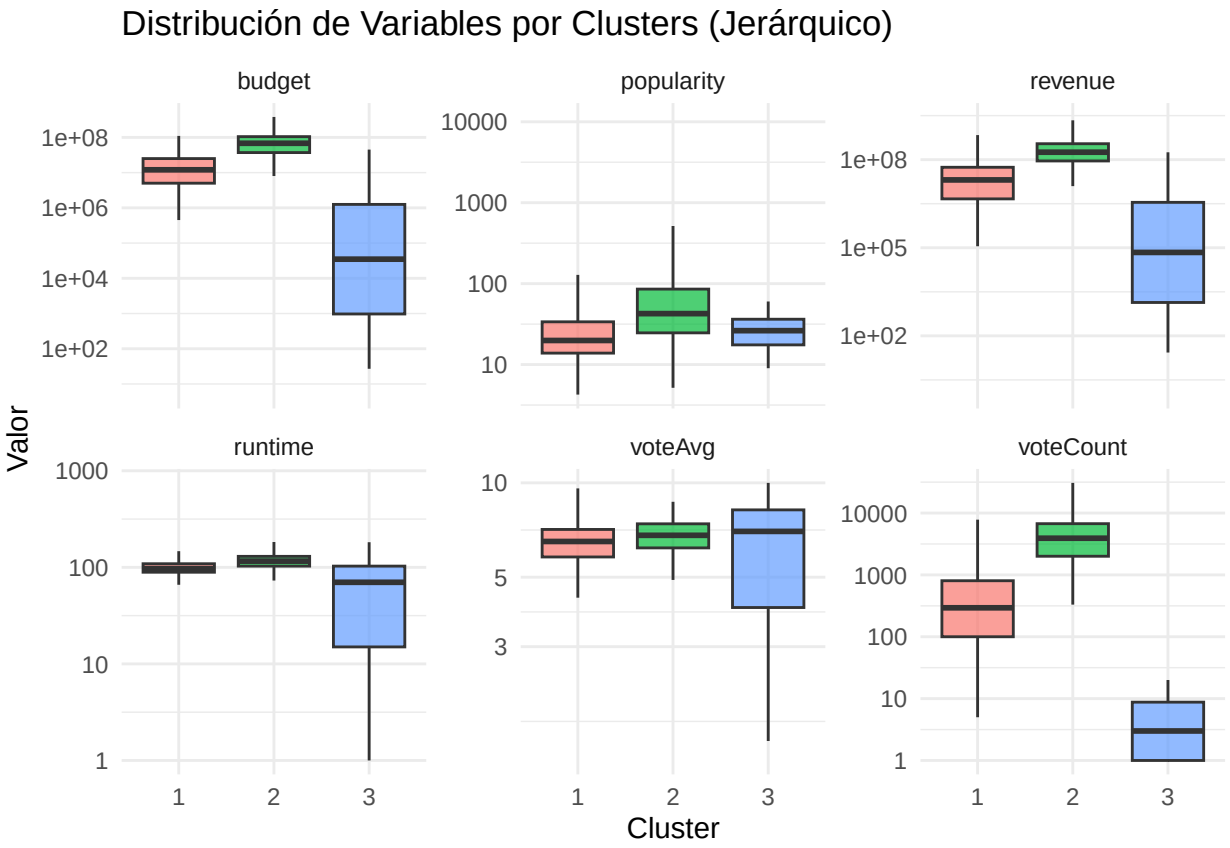


Clusters generados por K-Means:

```
El grupo 1: Presenta valores intermedios en presupuesto, recaudación y popularidad. Probablemente inclu
El grupo 2: Contiene películas con los presupuestos y recaudaciones más altos, alta popularidad y una g
El grupo 3: Se caracteriza por películas con presupuestos y recaudaciones bajas, menor popularidad y po
```

```
## Warning in scale_y_log10(): log-10 transformation introduced infinite values.
```

```
## Warning: Removed 9688 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Clusters generados por Clustering Jerárquico:

```
El grupo 1: Incluye películas con recaudación y presupuesto moderado,  popularidad mas bien baja.
El grupo 2: Peliculas de alto presupuesto, altas recaudacion y popularidad
El grupo 3: Agrupa peliculas de bajo presupuesto, con popularidades relativamente bajas y recaudaciones
```

3 Análisis de componentes Principales

Estudia la matriz de correlación, la agrega y explica lo que observa en ella

En el analisis se realiza a patir de una muestra de datos de 10000 peliculas obtenidos de la plataforma The movie DB. Se evalua la correlacion entre variables. Se presentan las varibales que se incuyen en el analisis:

Indice de popularidad de la película Presupuesto de la película Ingreso de la película Duración de la película Cantidad de géneros que representan la película Cantidad de companias productoras que participaron en la película Cantidad de paises que se llevó a cabo la pelicula Número de votos en la platadorma de la película Promedio de votos en la plataforma de la película Índice de popularidad del elenco de la película Cantidad de personas que actúan en la película Cantidad de actrices en el elenco de la película Cantidad de actores en el elenco de la película.

```
datos$castWomenAmount<- as.numeric(datos$castWomenAmount)
```

```
## Warning: NAs introduced by coercion
```

```r
datos$castMenAmount<- as.numeric(datos$castMenAmount)
```

```
## Warning: NAs introduced by coercion
```

```r
datos$actorsPopularity <-as.character(datos$actorsPopularity)
datos$actorsPopularity<- strsplit(datos$actorsPopularity, "\\|")
```

```
## Warning in strsplit(datos$actorsPopularity, "\\|"): unable to translate 'Self -
## President, Marvel Studios (archive footage)|Self - Director (archive
## footage)|Self - Executive Producer (archive footage)|Self - Supervising
## Producer, The Falcon and the Winter Soldier|Self - Supervising Producer,
## WandaVision|Self (archive foota...' to a wide string
```

```
## Warning in strsplit(datos$actorsPopularity, "\\|"): input string 9121 is
## invalid
```

```r
datos$actorsPopularity<-lapply(datos$actorsPopularity, as.numeric, use = "pairwise.complete.obs")
```

```
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
```

```
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
```

```
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
## Warning in lapply(datos$actorsPopularity, as.numeric, use =
## "pairwise.complete.obs"): NAs introduced by coercion
```

```
datos$actorsPopularity<- sapply(datos$actorsPopularity, function(x) if (all(is.na(x))) NA else mean(x, 
datos$actorsPopularity<- sapply(datos$actorsPopularity, function(x) {
  if (all(is.na(x))) {
    return(NA)  # Si todos son NA, el promedio es NA
  } else {
    return(mean(x, na.rm = TRUE))  # Calcular la media sin contar los NA
  }
})

sub_datos<-datos[, c("popularity", "budget", "revenue", "runtime", "genresAmount", "productionCoAmount"
matriz_cor <- cor(sub_datos, use = "pairwise.complete.obs" )
determinante<-det(matriz_cor)
```

La determinante es

```
print(determinante)
```

```
## [1] 0.08460687
```

indicando que las variabes estan relacionadas entre si

#Determina si es posible usar la técnica de análisis factorial para hallar las componentes principales

```
KMO(as.matrix(sub_datos))
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = as.matrix(sub_datos))
## Overall MSA =  0.67
## MSA for each item =
```

```
##                 popularity                     budget                    revenue
##                       0.81                       0.75                       0.68
##                    runtime              genresAmount        productionCoAmount
##                       0.72                       0.64                       0.52
## productionCountriesAmount                 voteCount                    voteAvg
##                       0.48                       0.75                       0.55
##           castWomenAmount             castMenAmount
##                       0.46                       0.51
```

El indice es de 0.67 lo cual es un valor regular, es suficiente pero no el ideal.

• Determina si vale la pena aplicar las componentes principales interpretando la prueba de esfericidad de Bartlett

```
cortest.bartlett(sub_datos)
```

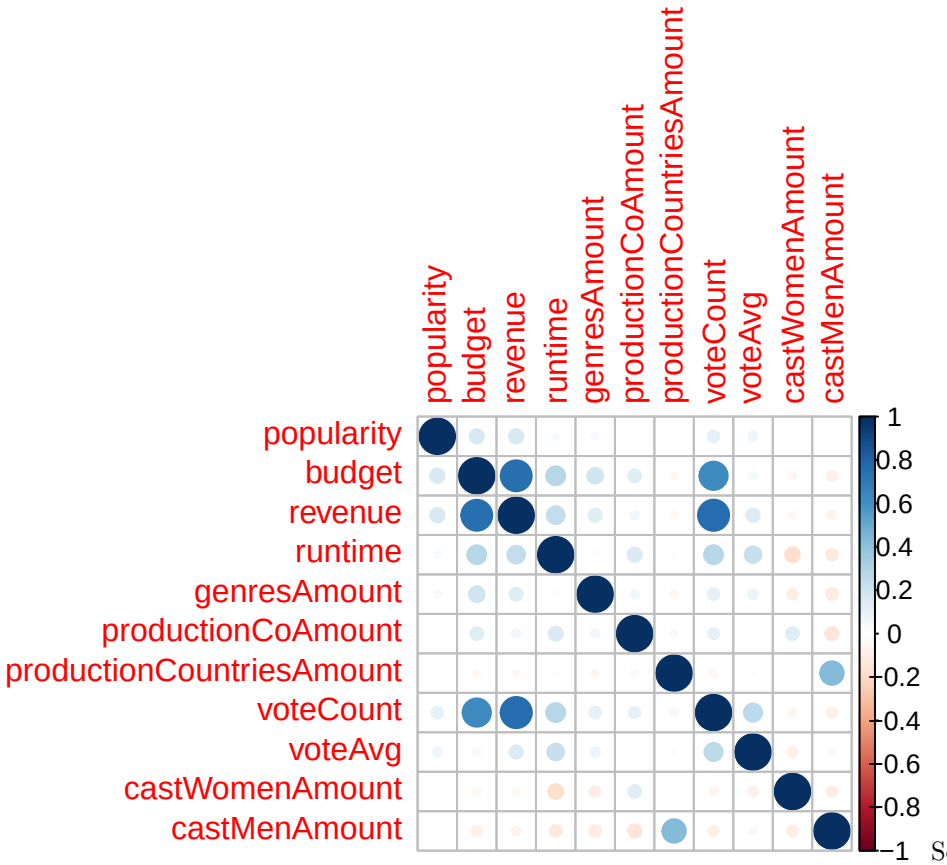
```
## R was not square, finding R from data
```

```
## $chisq
## [1] 24683.81
##
## $p.value
## [1] 0
##
## $df
## [1] 55
```

Como P=0 por lo que se rechaza la hipotesis nula, implicando que el analisis factorial es apropiado.

• Obtiene los componentes principales y explica cuántos seleccionará para explicar la mayor variabilidad posible.

```
matriz_cor <- cor(sub_datos, use = "pairwise.complete.obs" )
corrplot(matriz_cor)
```



Segun la tabla vemos que la variable del presupuesto de la pelicula se correlaciona con ingreso de pelicula, asi mismo con el numero de votos en la plataforma. El numero de votos en la plataforma con el ingreso de la pelicula tambien estan correlacionados. El numero de actores en el elenco de las peliculas esta relacionado con la cantidad de paises en los que se rodo la pelicula.

• Interpreta los coeficientes principales.

```
pca_result<-princomp(covmat=matriz_cor,use = "pairwise.complete.obs" )
```

```
## Warning: In princomp.default(covmat = matriz_cor, use = "pairwise.complete.obs") :
##  extra argument 'use' will be disregarded
```

```
#compPrinc<-prcomp(sub_datos, scale = T, use = "pairwise.complete.obs")
#compPrinc
summary(pca_result)
```

```
## Importance of components:
##                           Comp.1    Comp.2    Comp.3    Comp.4     Comp.5
## Standard deviation     1.6670613 1.2035413 1.0882368 1.0497615 0.99313055
## Proportion of Variance 0.2526448 0.1316829 0.1076599 0.1001817 0.08966439
## Cumulative Proportion  0.2526448 0.3843277 0.4919876 0.5921694 0.68183377
##                            Comp.6     Comp.7     Comp.8     Comp.9    Comp.10
## Standard deviation     0.98388184 0.93595208 0.80386875 0.71186565 0.56377525
## Proportion of Variance 0.08800213 0.07963694 0.05874591 0.04606843 0.02889478
## Cumulative Proportion  0.76983590 0.84947284 0.90821875 0.95428717 0.98318195
##                           Comp.11
## Standard deviation     0.43011460
## Proportion of Variance 0.01681805
## Cumulative Proportion  1.00000000
```

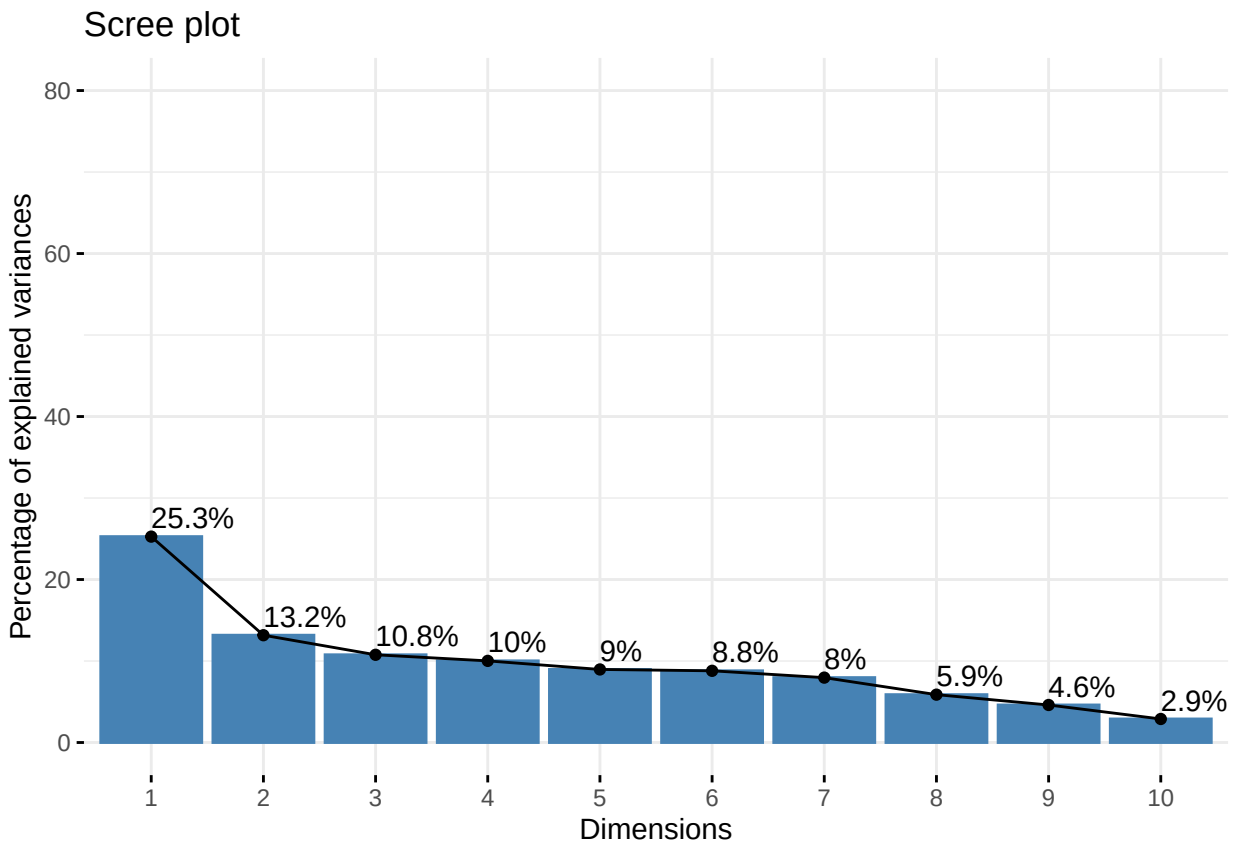
#regla de Kaiser

```
valores_propios<-pca_result$sdev^2
valores_propios
```

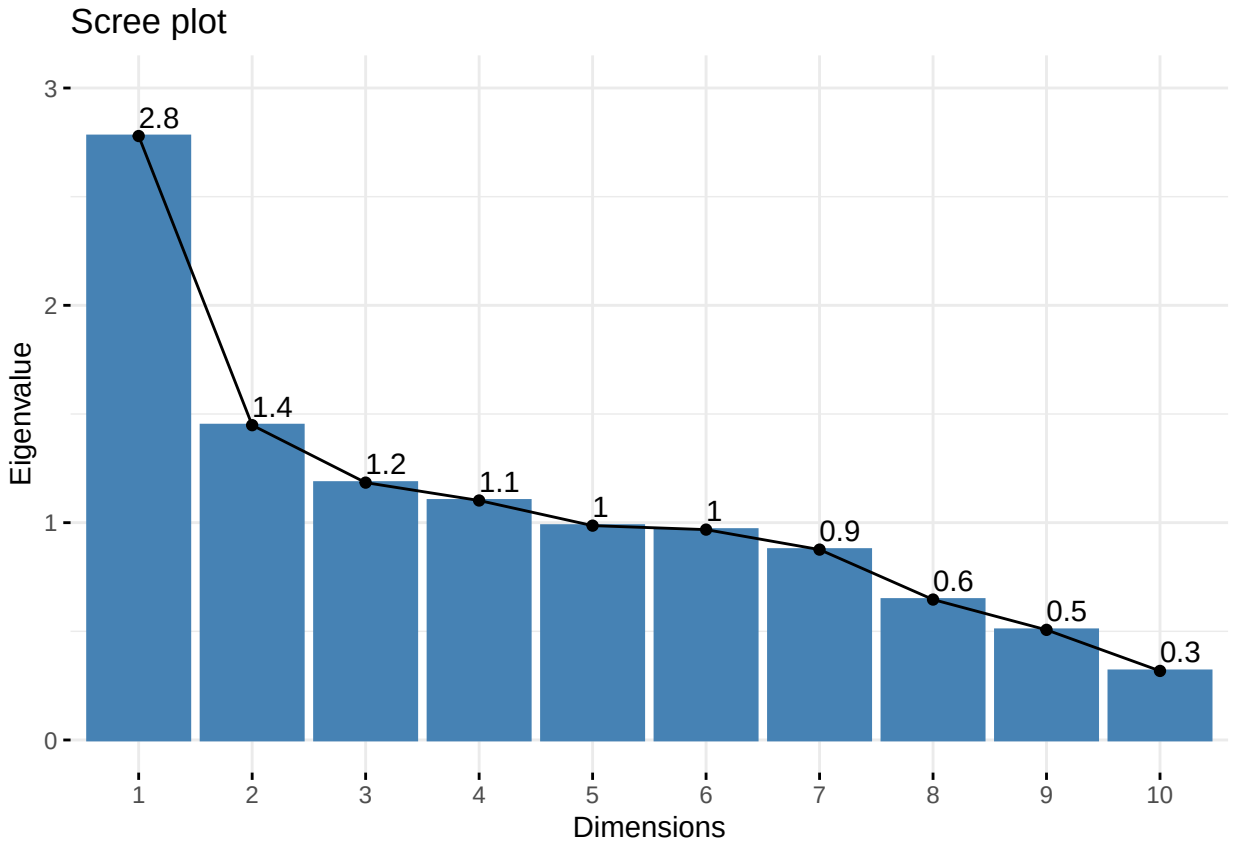
```
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7    Comp.8
## 2.7790932 1.4485116 1.1842592 1.1019991 0.9863083 0.9680235 0.8760063 0.6462050
##    Comp.9   Comp.10   Comp.11
## 0.5067527 0.3178425 0.1849986
```

Tomamos los 4 componentes principales.

```
fviz_eig(pca_result, addlabels = TRUE, ylim = c(0, 80))
```




```
fviz_eig(pca_result, addlabels = TRUE, choice = c("eigenvalue"), ylim = c(0, 3))
```



Interpretacion: EL componenete 1 se relaciona con el exito comercial y popularidad de la pelicula, las peliculas con alto presupuesto alto ingreso, y votaciones tienen valores altos en este componente.

El componente 2, puede indicar la cantidad de actores en el elenco pueden estar asociados con los paises productores.

El componente 3, inidica un mayor numero de mujeres en el elenco tienden a tener menor puntuacion.

El componente 4, Los valores altos podriamos relacionarlo con las peliculas independientes, es decir con menos productoras.

2. Reglas de Asociación

2.1. Obtenga reglas de asociación interesantes del conjunto de datos usando el algoritmo "A priori". Recuerde discretizar las variables numéricas. Genere reglas con diferentes niveles de confianza y soporte. Discuta los resultados. Si considera que debe eliminar variables

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

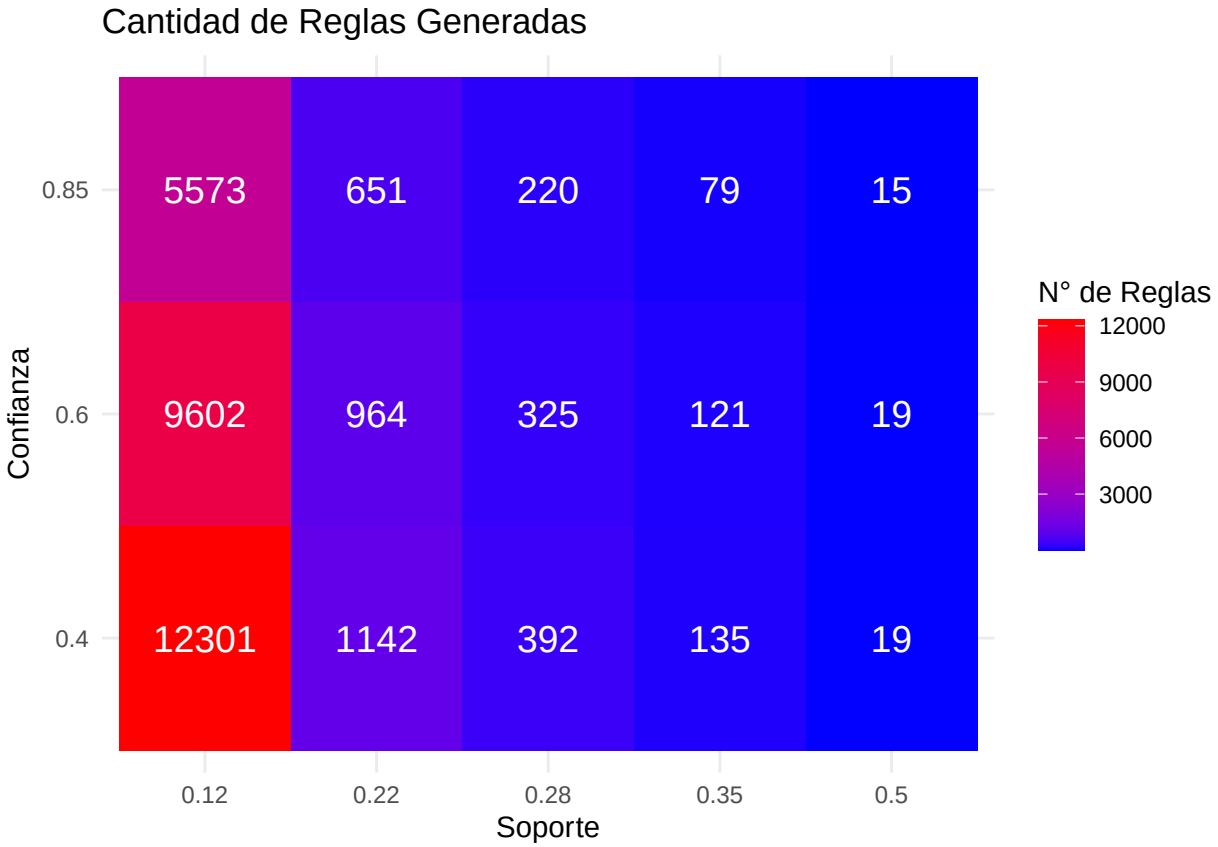
```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
##
## Attaching package: 'arules'
```

```
## The following object is masked from 'package:flexclust':
##
```

```
##     info

## The following object is masked from 'package:modeltools':
##
##     info

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

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

## Warning: There were 3 warnings in `mutate()`.
## The first warning was:
## i In argument: `across(where(is.numeric), discretize)`.
## Caused by warning:
## ! The calculated breaks are: 0, 0, 1.2e+07, 3.8e+08
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.

## Warning: Column(s) 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18,
## 19, 20, 21, 22, 23, 24, 25, 26, 27 not logical or factor. Applying default
## discretization (see '? discretizeDF').

## Warning in discretize(x = c(4000000L, 21000000L, 11000000L, 94000000L, 55000000L, : The calculated bi
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.

## Warning in discretize(x = c(4257354, 12136938, 775398007, 940335536, 677387716, : The calculated brea
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.

## Warning in discretize(x = c(1L, 2L, 1L, 1L, 1L, 1L, 1L, 12L, 1L, 1L, 1L, : The calculated breaks are
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.
```

```r
library(ggplot2)

# Crear el gráfico de calor
ggplot(resultados, aes(x = factor(Soporte), y = factor(Confianza), fill = Num_Reglas)) +
  geom_tile() +
  geom_text(aes(label = Num_Reglas), color = "white", size = 5) +
  scale_fill_gradient(low = "blue", high = "red") +
  labs(title = "Cantidad de Reglas Generadas",
       x = "Soporte",
       y = "Confianza",
       fill = "N° de Reglas") +
  theme_minimal()
```



Con el mapa de calor de el numero de reglas tenemos una idea de que soporte y confianza es buena idea elegir.Para discutir sobre las variables que mas importan para las reglas y las que menos vamos a elegir un soporte de 30% y una confianza de 60%

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##        0.85    0.1    1 none FALSE            TRUE       5     0.5      3
##  maxlen target  ext
##       7  rules TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 5000
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[67533 item(s), 10000 transaction(s)] done [0.03s].
## sorting and recoding items ... [7 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [15 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

Sorpresa, a la base de datos le agregue las etiquetas de los clusters. La mayoria de regras tienen algo que ver con el cluster jerárquico 1 . Pero las reglas tienen intervalos muy grandes, tanto que no aportan conocimineto

relevante Yo quitaría esa variable.

```
variables <- c("budget","genres","homePage","productionCompany","productionCompanyCountry","productionC
```

```
pelis <- datos %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.numeric), discretize))
```

```
## Warning: There were 3 warnings in `mutate()`.
## The first warning was:
## i In argument: `across(where(is.numeric), discretize)`.
## Caused by warning:
## ! The calculated breaks are: 0, 0, 1.2e+07, 3.8e+08
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

```
transa <- as(datos,"transactions")
```

```
## Warning: Column(s) 1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14, 15, 16, 17, 18,
## 19, 20, 21, 22, 23, 24, 25, 26, 27 not logical or factor. Applying default
## discretization (see '? discretizeDF').
```

```
## Warning in discretize(x = c(4000000L, 21000000L, 11000000L, 94000000L, 55000000L, : The calculated b
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.
```

```
## Warning in discretize(x = c(4257354, 12136938, 775398007, 940335536, 677387716, : The calculated brea
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.
```

```
## Warning in discretize(x = c(1L, 2L, 1L, 1L, 1L, 1L, 1L, 12L, 1L, 1L, 1L, : The calculated breaks are
##   Only unique breaks are used reducing the number of intervals. Look at ? discretize for details.
```

```
reglas<-apriori(transa, parameter = list(support= 0.20, target="frequent", minlen=2, maxlen=4))
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5     0.2      2
##  maxlen            target  ext
##       4 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2000
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[67533 item(s), 10000 transaction(s)] done [0.03s].
## sorting and recoding items ... [42 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4
```

```
## Warning in apriori(transa, parameter = list(support = 0.2, target = "frequent",
## : Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!
```

```
##  done [0.01s].
## sorting transactions ... done [0.00s].
## writing ... [707 set(s)] done [0.00s].
```

```
## creating S4 object  ... done [0.00s].
```

```
#inspect(sort(reglas))
```

- Prueba con varios valores de confianza y soporte, y decide si quitar o no características para obtener mejores hallazgos.
- Discute sobre las reglas de asociación más interesantes teniendo en cuenta sus niveles de confianza y soporte

```
reglas<-apriori(transa, parameter = list(support= 0.30, confidence=0.7, target="frequent", minlen=2, ma
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5     0.3      2
##  maxlen            target  ext
##       4 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 3000
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[67533 item(s), 10000 transaction(s)] done [0.03s].
## sorting and recoding items ... [39 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4
```

```
## Warning in apriori(transa, parameter = list(support = 0.3, confidence = 0.7, :
## Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!
```

```
##  done [0.01s].
## sorting transactions ... done [0.00s].
## writing ... [138 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(reglas), 5))
```

```
##     items                                 support count
## [1] {productionCountriesAmount=[1,155],
##      Cluster_Jerarquico=1}                 0.8314  8314
## [2] {originalLanguage=en,
##      productionCountriesAmount=[1,155]}    0.7720  7720
## [3] {revenue=[0,2.03e+07),
##      productionCountriesAmount=[1,155]}    0.6579  6579
## [4] {budget=[0,1.2e+07),
##      productionCountriesAmount=[1,155]}    0.6560  6560
## [5] {revenue=[0,2.03e+07),
##      Cluster_Jerarquico=1}                 0.6539  6539
```

Con un soporte alto, de mayor o igual a 0.30, por lo que se tomaron en cuenta solo las reglas donde aparecen al menos el 30% de las transacciones. 1. Notese que se observan muchas, video=FALSE, aparece en casi toda la lista por lo que no seria demasiado util y se podria eliminar en proximas pruebas. 2. originialLanguage=en lo cual indica que la mayoria de las peliculas entan en ingles, si el 70% de estos datos tiene esta caracteristica se recomienda eliminar de para proximas pruebas. 3. Las reglas de productionCountry=United States of

America, originalLanguage=en indicando que la mayoria de peliculas provenientes de Estados Unidos esten en ingles, seria util probar con valores mas bajos. 4. La regla budget=0, revenue=0, sugiere que hay muchas peliculas con presupuesto no registrados muy bajos.

```
reglas<-apriori(transa, parameter = list(support= 0.20, confidence=0.6, target="frequent", minlen=2, ma
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5     0.2      2
##  maxlen            target  ext
##       4 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 2000
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[67533 item(s), 10000 transaction(s)] done [0.04s].
## sorting and recoding items ... [42 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4
```

```
## Warning in apriori(transa, parameter = list(support = 0.2, confidence = 0.6, :
## Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!
```

```
##  done [0.01s].
## sorting transactions ... done [0.00s].
## writing ... [707 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(reglas), 5))
```

```
##     items                                 support count
## [1] {productionCountriesAmount=[1,155],
##      Cluster_Jerarquico=1}                 0.8314  8314
## [2] {originalLanguage=en,
##      productionCountriesAmount=[1,155]}    0.7720  7720
## [3] {revenue=[0,2.03e+07),
##      productionCountriesAmount=[1,155]}    0.6579  6579
## [4] {budget=[0,1.2e+07),
##      productionCountriesAmount=[1,155]}    0.6560  6560
## [5] {revenue=[0,2.03e+07),
##      Cluster_Jerarquico=1}                 0.6539  6539
```

1. las reglas> video=FALSE, originalLanguage=en, sigue dominando hay mas reglas relacionadas con presupuesto y revenue
2. Aun aparece la regla budget=0, revenue=0 lo que sugiere la existencia de un grupo de peliculas con poco presupuesto

```
reglas<-apriori(transa, parameter = list(support= 0.10, confidence=0.5, target="frequent", minlen=2, ma
```

```
## Apriori
##
## Parameter specification:
```

```
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##          NA    0.1    1 none FALSE            TRUE       5     0.1      2
##  maxlen            target  ext
##       4 frequent itemsets TRUE
##
## Algorithmic control:
##  filter tree heap memopt load sort verbose
##     0.1 TRUE TRUE  FALSE TRUE    2    TRUE
##
## Absolute minimum support count: 1000
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[67533 item(s), 10000 transaction(s)] done [0.04s].
## sorting and recoding items ... [45 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4
```

```
## Warning in apriori(transa, parameter = list(support = 0.1, confidence = 0.5, :
## Mining stopped (maxlen reached). Only patterns up to a length of 4 returned!
```

```
##  done [0.03s].
## sorting transactions ... done [0.00s].
## writing ... [5036 set(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
inspect(head(sort(reglas), 5))
```

```
##     items                                  support count
## [1] {productionCountriesAmount=[1,155],
##      Cluster_Jerarquico=1}                  0.8314  8314
## [2] {originalLanguage=en,
##      productionCountriesAmount=[1,155]}     0.7720  7720
## [3] {revenue=[0,2.03e+07),
##      productionCountriesAmount=[1,155]}     0.6579  6579
## [4] {budget=[0,1.2e+07),
##      productionCountriesAmount=[1,155]}     0.6560  6560
## [5] {revenue=[0,2.03e+07),
##      Cluster_Jerarquico=1}                  0.6539  6539
```

1. Las reglas: video=FALSE, originalLanguage=en, sigue dominando en los items
2. Los items budget=0, revenue=0 son de mayor frecuencia, por lo que los datos tienen valores de presupuesto e ingresos en cero

```
quality(reglas)$lift <- interestMeasure(reglas, measure = "lift", transa =transa)
inspect(head(sort(reglas, by ="lift"), n=10))
```

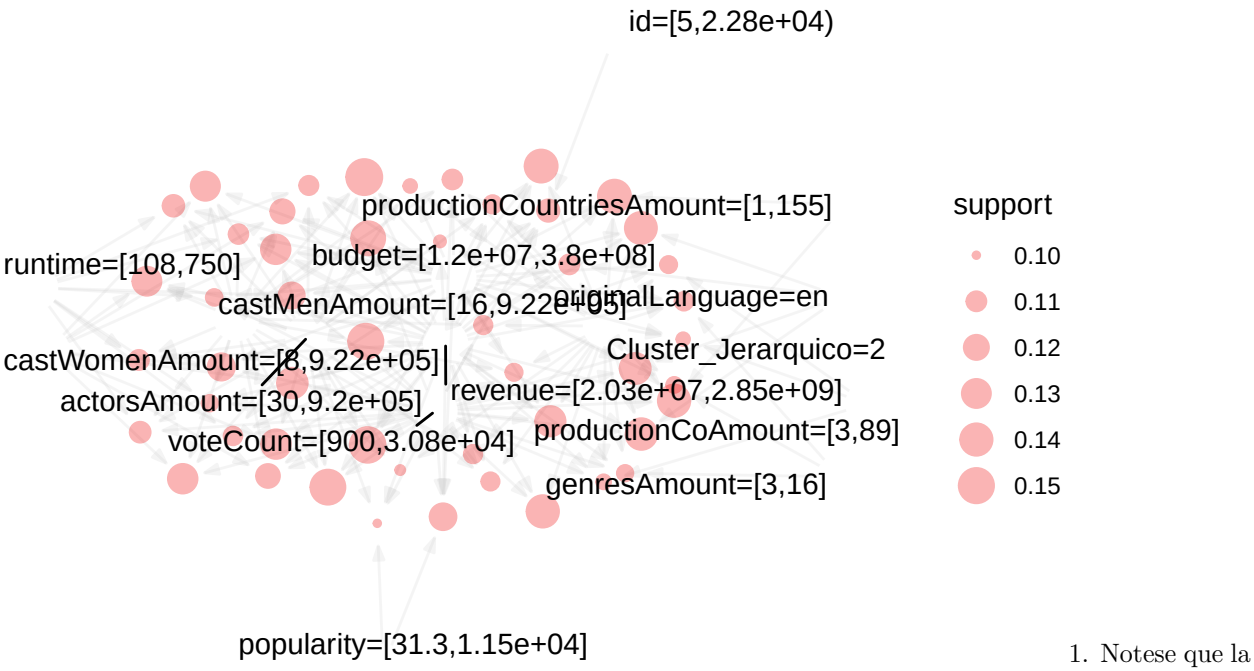
```
##      items                             support count     lift
## [1]  {budget=[1.2e+07,3.8e+08],
##       revenue=[2.03e+07,2.85e+09],
##       voteCount=[900,3.08e+04],
##       Cluster_Jerarquico=2}             0.1336  1336 22.61583
## [2]  {revenue=[2.03e+07,2.85e+09],
##       voteCount=[900,3.08e+04],
##       actorsAmount=[30,9.2e+05],
##       Cluster_Jerarquico=2}             0.1074  1074 18.13202
## [3]  {budget=[1.2e+07,3.8e+08],
##       revenue=[2.03e+07,2.85e+09],
```

```
##       actorsAmount=[30,9.2e+05],
##       Cluster_Jerarquico=2}          0.1076  1076 18.06284
## [4]  {budget=[1.2e+07,3.8e+08],
##       voteCount=[900,3.08e+04],
##       actorsAmount=[30,9.2e+05],
##       Cluster_Jerarquico=2}          0.1061  1061 17.81104
## [5]  {budget=[1.2e+07,3.8e+08],
##       revenue=[2.03e+07,2.85e+09],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1097  1097 17.72473
## [6]  {revenue=[2.03e+07,2.85e+09],
##       voteCount=[900,3.08e+04],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1077  1077 17.50075
## [7]  {budget=[1.2e+07,3.8e+08],
##       voteCount=[900,3.08e+04],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1075  1075 17.36927
## [8]  {budget=[1.2e+07,3.8e+08],
##       actorsAmount=[30,9.2e+05],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1025  1025 16.42346
## [9]  {revenue=[2.03e+07,2.85e+09],
##       actorsAmount=[30,9.2e+05],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1013  1013 16.32369
## [10] {voteCount=[900,3.08e+04],
##       actorsAmount=[30,9.2e+05],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1004  1004 16.17866
```

```
plot(head(sort(reglas, by="lift"), n=50), method = "graph", control=list(cex=.8))
```

```
## Warning: Unknown control parameters: cex
```

```
## Available control parameters (with default values):
## layout    =  stress
## circular  =  FALSE
## ggraphdots    =  NULL
## edges     =  <environment>
## nodes     =  <environment>
## nodetext  =  <environment>
## colors    =  c("#EE0000FF", "#EEEEEEFF")
## engine    =  ggplot2
## max   =  100
## verbose   =  FALSE
```



1. Notese que la regla productionCountry=United States of America, asociado con originalLanguage=en, sugiriendo que la mayoria de peliculas de Estados Unidos tiene en ingles como idioma original y son producidas en un solo pais 2. Las reglas de budget=0, revenue=0, indicio de peliculas sin datos de presupuesto e ingresos registrados 3. genresAmount=2, genresAmount=3, aparecen conectados, posible tendencia de peliculas con numero bajo de generos

(25 puntos) Hallazgos y conclusiones.

- Hace un resumen de los hallazgos que arrojó el agrupamiento.

El análisis de agrupamiento reveló tres grupos principales en el conjunto de datos de películas:

- Grupo 1: Películas con recaudación y presupuesto moderado, y una popularidad más bien baja.
- Grupo 2: Películas de alto presupuesto, alta recaudación y alta popularidad. Este grupo probablemente incluye grandes producciones o éxitos de taquilla.
- Grupo 3: Películas de bajo presupuesto con popularidades relativamente bajas y recaudaciones muy dispersas. Este grupo parece representar películas independientes o de nicho.

El método de silueta mostró que el clustering jerárquico obtuvo mejores resultados que k-medias, con una silueta promedio de 0.35 frente a 0.15, lo que indica que las agrupaciones realizadas por el clustering jerárquico son más compactas y mejor definidas.

- Llega a conclusiones sobre el análisis de componentes principales

La componente 1 nos indica el exito comercual y popularidad. Relacionado con el presupuesto, ingreso y votaciones de la pelicula, la cua indica que si una pelicula tiene valores altos en este componente, tiende a ser mas exitosa y populares generando mas ingresos

El componente 2, se relaciona con la cantidad de actores que hay en el elenco y los paises productores

El componente 3 acentua la presencia femenina en el elenco y puntuacion, sugiendo que las peliculas con mayor numero de actrices tienden a tener menor puntuacion. Podria refejar prejuicios o esteriotipos en la industria o en el publico

El componente 4 relaciona las peliculas independientes que requieren de mas evidencia para llegar a una conclusion, es decir analizar mas variable.

- Determina las reglas de asociación más interesantes.

```r
quality(reglas)$lift <- interestMeasure(reglas, measure = "lift", transa =transa)
inspect(head(sort(reglas, by ="lift"), n=10))
```

```
##      items                           support count     lift
## [1]  {budget=[1.2e+07,3.8e+08],
##       revenue=[2.03e+07,2.85e+09],
##       voteCount=[900,3.08e+04],
##       Cluster_Jerarquico=2}          0.1336  1336 22.61583
## [2]  {revenue=[2.03e+07,2.85e+09],
##       voteCount=[900,3.08e+04],
##       actorsAmount=[30,9.2e+05],
##       Cluster_Jerarquico=2}          0.1074  1074 18.13202
## [3]  {budget=[1.2e+07,3.8e+08],
##       revenue=[2.03e+07,2.85e+09],
##       actorsAmount=[30,9.2e+05],
##       Cluster_Jerarquico=2}          0.1076  1076 18.06284
## [4]  {budget=[1.2e+07,3.8e+08],
##       voteCount=[900,3.08e+04],
##       actorsAmount=[30,9.2e+05],
##       Cluster_Jerarquico=2}          0.1061  1061 17.81104
## [5]  {budget=[1.2e+07,3.8e+08],
##       revenue=[2.03e+07,2.85e+09],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1097  1097 17.72473
## [6]  {revenue=[2.03e+07,2.85e+09],
##       voteCount=[900,3.08e+04],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1077  1077 17.50075
## [7]  {budget=[1.2e+07,3.8e+08],
##       voteCount=[900,3.08e+04],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1075  1075 17.36927
## [8]  {budget=[1.2e+07,3.8e+08],
##       actorsAmount=[30,9.2e+05],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1025  1025 16.42346
## [9]  {revenue=[2.03e+07,2.85e+09],
##       actorsAmount=[30,9.2e+05],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1013  1013 16.32369
## [10] {voteCount=[900,3.08e+04],
##       actorsAmount=[30,9.2e+05],
##       castMenAmount=[16,9.22e+05],
##       Cluster_Jerarquico=2}          0.1004  1004 16.17866
```

La regla mas fuerte (lift=2.94): Si la pelicula tiene presupuesto 0 budget=0 y no genero ingresos revenue=0, es probable que sea producida desde una sola compania productionCoAmount=1 en un solo pais productionCountry=1. Posibles proyectos independientes

Segunda regla (lift=2.86) {id=[3.77e+05,9.22e+05],budget=0, revenue=0. Rango de peliculas con cero presupuesto y cero ingreso, lo que pueden ser proyectos cancelados,

Tercera regla la mayoria de peliculas con cero presupuesto y cero ingreso, estan en ingles.

- Propone sugerencias a CineVision Studios para nuevos desarrollos y mejora de áreas teniendo en cuenta los descubrimientos que hizo.

Varias de las reglas con presupuesto=0 e ingresos=0, se sugiere conciderar invertir en proyectos independientes con mejor financiamiento y buen promocional para mas exito.

Identificar factores que diferencian producciones de bajo presupuesto y gran exito para replica

Varias reglas detectaoron que mucas peliculas son producidas por una sola compania y en un solo pais, predominando EEUU y que son en ingles. Se recomienda alianzas entre estudios para diversificar producciones, Abrirse nuevos mercados, aportando en la diversidad cultural, implicando conocimiento global.

Al tener varios datos de forma budget=0 y revenue=0 podria indicar problemas de registro de datos, por lo que se recomienda mejorar la recopilacion de datos para tener informacion mas precisas

Desarollar mejores estrategias de marketing mas efectivas para aprovechar maxima difucion.