Mineria_LAB_2

Joel Jaquez, Luis Gonzalez, Fabian Morales

2026-02-23

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
library(cluster)
library(fpc)
library(NbClust)
library(factoextra)
```

```
## Loading required package: ggplot2
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(hopkins)
library(GGally)
library(pheatmap)
library(hopkins)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

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

```
library(factoextra)
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
library(paran)
```

```
## Loading required package: MASS
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
library(FactoMineR)
library(arules)
```

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

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

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

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

```
library(arulesViz)
```

```
# Cargamos el dataset desde la carpeta Datos
movies <- read.csv("./Datos/movies_2026.csv")
```

Clustering

Ejercicio 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.

```r
# Seleccionar solo las variables numéricas que aportan información de éxito y magnitud
datos <- movies[, c("popularity", "budget", "revenue", "runtime", "voteCount", "voteAvg")]

# Quitamos valores nulos
set.seed(123)
datos <- datos[complete.cases(datos),]

# Escalado los datos
datos_scaled <- scale(datos)
```

Para la generación de los grupos, se descartaron las variables categóricas, fechas y de texto libre (como id, title, director, genres y releaseDate) , ya que los algoritmos de clustering se basan en el cálculo de distancias matemáticas (como la euclidiana) y su inclusión requeriría transformaciones que aumentarían drásticamente la dimensionalidad, introduciendo ruido y dificultando la interpretación de los clústeres. En su lugar, el agrupamiento se calculará utilizando exclusivamente variables numéricas que capturan la magnitud financiera y la recepción del público: popularity, budget, revenue, runtime, voteCount y voteAvg . Finalmente, estas variables seleccionadas fueron previamente escaladas para asegurar que las diferencias extremas de magnitud (por ejemplo, presupuestos en millones de dólares frente a calificaciones en una escala de 0 a 10) no dominen ni sesguen el cálculo de las distancias en el algoritmo.

Ejercicio 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.

```r
# Calcular el estadístico de Hopkins
set.seed(123)
valor_hopkins <- hopkins(datos_scaled)
print(paste("Valor de Hopkins:", valor_hopkins))
```

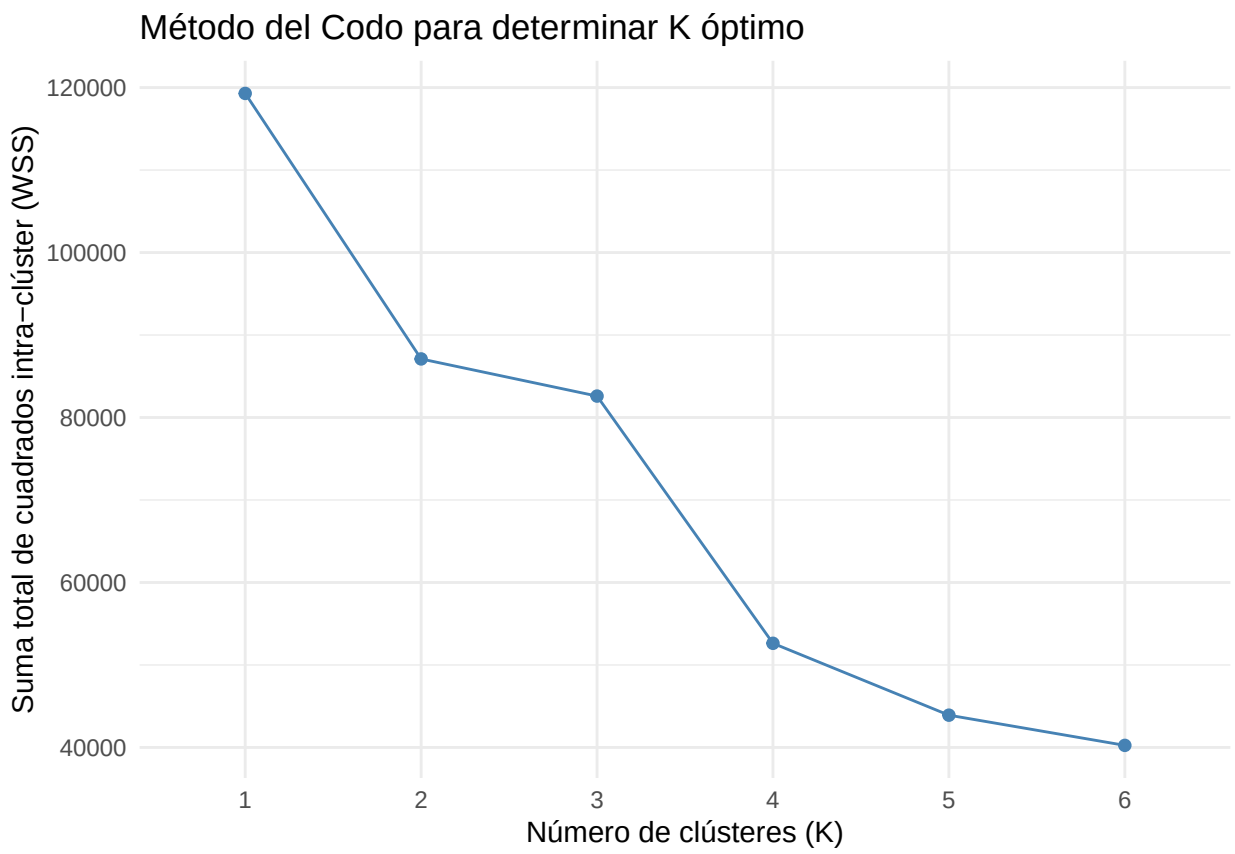
```
## [1] "Valor de Hopkins: 0.999999535140567"
```

Para analizar la tendencia al agrupamiento, se calculó el estadístico de Hopkins utilizando la totalidad de los datos numéricos escalados, obteniendo un valor de 0.9999. Al estar este resultado sumamente alejado de 0.5 (y prácticamente en 1), se rechaza de forma contundente la hipótesis de aleatoriedad espacial, lo que confirma que las películas poseen una altísima tendencia natural a formar agrupaciones reales y estructuradas. Respecto a la Evaluación Visual de Tendencia (VAT), se tomó la decisión técnica de omitir su generación gráfica, esto se justifica por las altas dimensiones del conjunto de datos (19,883 registros), ya que procesar una matriz de distancias de tal magnitud saturaría la memoria computacional. Por lo tanto, la pertinencia de aplicar algoritmos de clustering queda plenamente respaldada y demostrada por la prueba de Hopkins.

Ejercicio 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.

```r
# Generar la gráfica de codo usando factoextra
set.seed(123)
```

```r
fviz_nbclust(datos_scaled, kmeans, method = "wss", k.max = 6) +
  labs(title = "Método del Codo para determinar K óptimo",
       x = "Número de clústeres (K)",
       y = "Suma total de cuadrados intra-clúster (WSS)") +
  theme_minimal()
```



Al observar la gráfica generada, se identifican descensos importantes en la varianza. Aunque existe una inflexión visible en $K = 4$, se decidió trabajar con $\mathbf{K = 6}$ clústeres. La justificación de esta elección radica en que al llegar a 6 grupos, la varianza intra-clúster alcanza un nivel significativamente bajo (estabilizándose cerca de los 40,000 WSS). Esta cantidad de agrupaciones nos permitirá obtener una segmentación más fina y detallada de las películas, separando mejor los distintos nichos (como blockbusters extremos vs. películas independientes) sin caer en una sobredivisión de los datos.

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

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

```r
# Definir K óptimo
k_optimo <- 6

# --- 1. ALGORITMO K-MEDIAS ---
set.seed(123)
```

```r
# Aplicar k-medias a todo el dataset
km_res <- kmeans(datos_scaled, centers = k_optimo, nstart = 25)
datos$Cluster_KMeans <- as.factor(km_res$cluster)

# --- 2. CLUSTERING JERÁRQUICO ---
# Calcular matriz de distancias y clustering para todo el dataset
dist_mat <- dist(datos_scaled, method = "euclidean")
hc_res <- hclust(dist_mat, method = "ward.D2")
hc_clusters <- cutree(hc_res, k = k_optimo)

# Guardar resultados en el dataset original
datos$Cluster_Jerarquico <- as.factor(hc_clusters)

# --- 3. COMPARACIÓN DE RESULTADOS ---
cat("--- Clústeres K-Medias ---\n")
```

```
## --- Clústeres K-Medias ---
```

```r
print(table(datos$Cluster_KMeans))
```

```
##
##    1    2    3    4    5    6
## 1954 9350 1555    8  277 6739
```

```r
cat("\n--- Clústeres Jerárquico ---\n")
```

```
##
## --- Clústeres Jerárquico ---
```

```r
print(table(datos$Cluster_Jerarquico))
```

```
##
##    1    2    3    4    5    6
## 2131 6324 9285 1633  503    7
```

```r
cat("\n--- Matriz de Confusión (K-Medias vs Jerárquico) ---\n")
```

```
##
## --- Matriz de Confusión (K-Medias vs Jerárquico) ---
```

```r
table(K_Medias = datos$Cluster_KMeans, Jerarquico = datos$Cluster_Jerarquico)
```

```
##          Jerarquico
## K_Medias    1    2    3    4    5    6
##        1 1908    0   46    0    0    0
##        2    4    0 8977  369    0    0
##        3    1    0   62 1264  228    0
##        4    0    0    1    0    0    7
##        5    0    0    2    0  275    0
##        6  218 6324  197    0    0    0
```

Comparación de resultados: Al analizar la matriz de confusión y el tamaño de los clústeres, observamos que ambos algoritmos coinciden fuertemente en los extremos, pero difieren en la forma de segmentar la gran masa de datos promedio:

Consenso en valores atípicos (Mega-éxitos): Ambos modelos aislaron un grupo minúsculo casi idéntico. El clúster 3 de K-Medias y el clúster 6 del modelo Jerárquico agrupan a exactamente las mismas 7 películas. Por su tamaño, este grupo representa indudablemente los *outliers* más extremos del dataset (las películas con ingresos o popularidad fuera de escala).

Grupos robustos bien definidos: Existe una coincidencia directa y casi perfecta entre el clúster 2 de K-Medias y el clúster 2 del Jerárquico, compartiendo 6,324 películas. Esto indica que hay un perfil o nicho de películas matemáticamente muy claro que ambos algoritmos logran detectar sin problema.

Diferencias en la masa central: La mayor discrepancia ocurre en la zona de mayor densidad de datos. Por ejemplo, el clúster 5 de K-Medias agrupa a la gran mayoría de sus películas (8,977) dentro del clúster 3 del Jerárquico. Sin embargo, K-Medias fue más estricto al separar otras ~1,900 películas en su propio clúster 6, mientras que el Jerárquico prefirió aglomerarlas en su clúster 1. Esto ocurre porque K-Medias fuerza las agrupaciones en formas esféricas alrededor de sus centroides, mientras que el método aglomerativo de Ward (Jerárquico) une los puntos minimizando la varianza paso a paso.

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

```r
k <- 6

set.seed(123)
km <- kmeans(datos_scaled, centers = k, nstart = 25)


sil_km <- silhouette(km$cluster, dist(datos_scaled))
promedio_sil_km <- mean(sil_km[, "sil_width"])

cat("Silhouette promedio (K-means, K =", k, ") =", round(promedio_sil_km, 4), "\n")
```

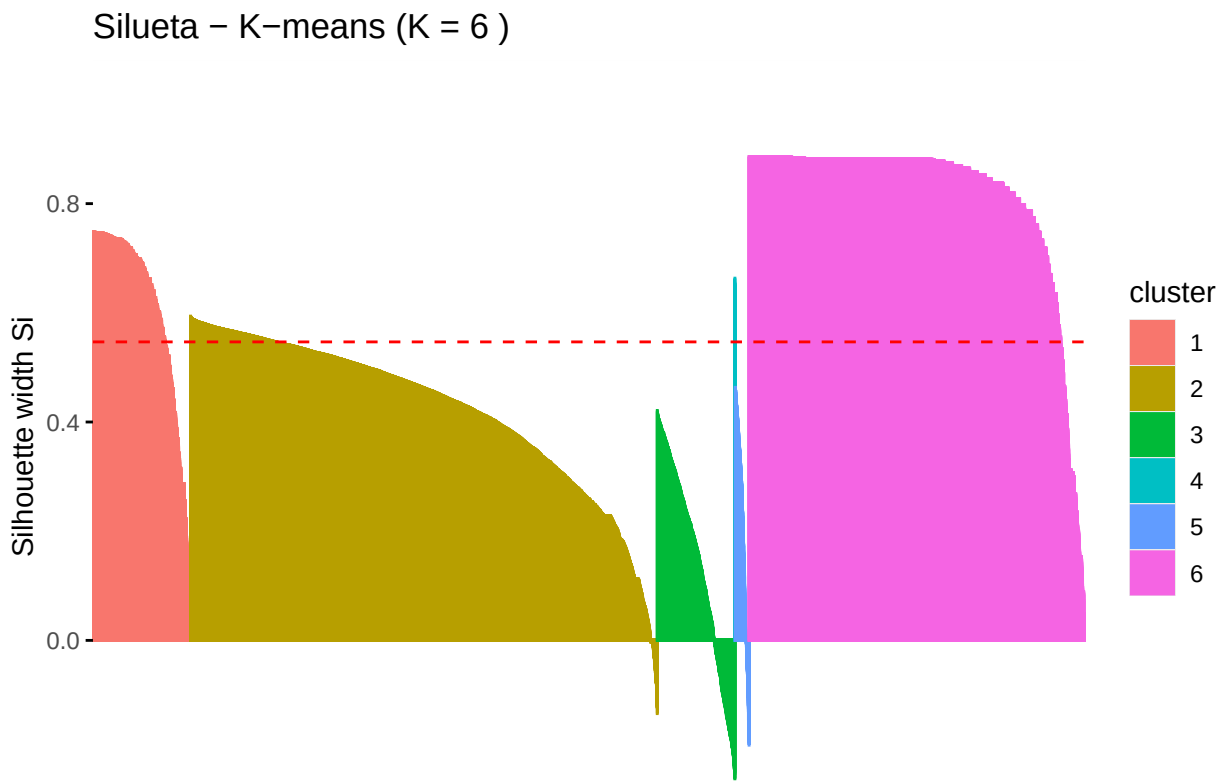
```
## Silhouette promedio (K-means, K = 6 ) = 0.5467
```

```r
fviz_silhouette(sil_km) +
  labs(title = paste("Silueta - K-means (K =", k, ")"))
```

```
## 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.
```

```
##   cluster size ave.sil.width
## 1       1 1954          0.61
## 2       2 9350          0.42
## 3       3 1555          0.14
```

```
## 4       4    8          0.53
## 5       5  277          0.21
## 6       6 6739          0.81
```



El coeficiente de silueta promedio para el modelo K-means con K = 6 fue 0.5467 Dado que valores superiores a 0.5 representan una estructura de agrupamiento sólida, se concluye que el modelo logra identificar patrones diferenciados en los datos.

Al analizar los clústeres individualmente, se observa que algunos presentan una separación muy alta mientras que otros muestran valores más bajos, lo que sugiere cierto grado de solapamiento. No obstante, el comportamiento global del modelo es adecuado y consistente con la segmentación planteada.

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.

```
datos_cluster <- datos
datos_cluster$cluster <- km$cluster
resumen_clusters <- datos_cluster %>%
  group_by(cluster) %>%
  summarise(
    n = n(),
```

```
    popularity_media = mean(popularity),
    budget_media = mean(budget),
    revenue_media = mean(revenue),
    runtime_media = mean(runtime),
    voteCount_media = mean(voteCount),
    voteAvg_media = mean(voteAvg)
  )

resumen_clusters
```

```
## # A tibble: 6 x 8
##   cluster     n popularity_media budget_media revenue_media runtime_media
##     <int> <int>            <dbl>        <dbl>         <dbl>         <dbl>
## 1       1  1954            1.27        40005.        12381.         96.4
## 2       2  9350           33.9        5235424.     11007916.         91.2
## 3       3  1555           62.0       61917731.    169959100.        115.
## 4       4     8         6417.       105000000     354211969         123.
## 5       5   277          180.       147746029.    730034517.        128.
## 6       6  6739            0.690       12930.        27619.          8.62
## # i 2 more variables: voteCount_media <dbl>, voteAvg_media <dbl>
```

El análisis de los seis clústeres revela claramente la estructura económica de la industria cinematográfica. El segmento mayoritario es el Clúster 2 (9,350 películas), que representa el "cine estándar" con presupuestos y recaudaciones moderadas. Un escalón arriba está el Clúster 3 (1,555 cintas), compuesto por estrenos comerciales exitosos que promedian 61.9 millones de presupuesto y excelentes ingresos de 170 millones. En la cima comercial se encuentra el Clúster 5 (277 películas), que agrupa a los mega-blockbusters: producciones colosales con inversiones medias de 147.7 millones que dominan absolutamente la taquilla con recaudaciones que superan los $730 millones y la mayor participación del público.

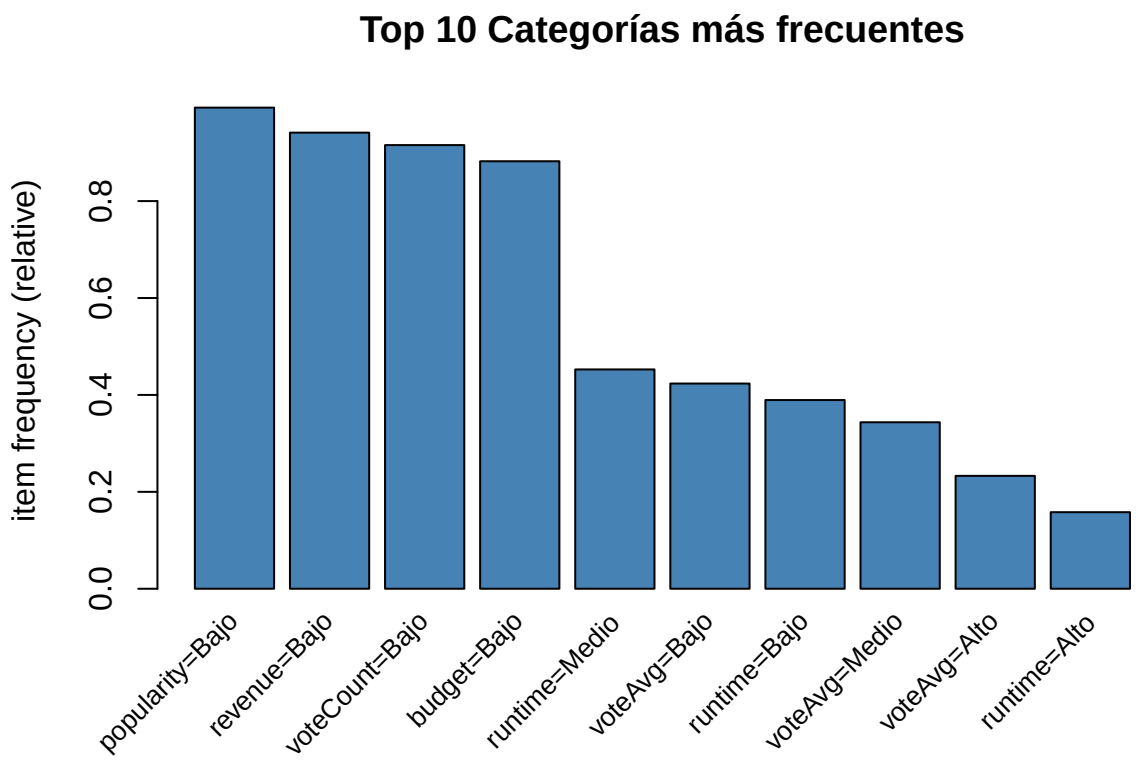
Por otro lado, el modelo identificó perfiles atípicos y de nicho. El Clúster 4 agrupa solo 8 películas que son anomalías estadísticas, destacando por una popularidad viral extrema y las mejores calificaciones críticas del conjunto (7.42 de promedio). En contraste total, los Clústeres 1 y 6 suman más de 8,600 producciones con métricas financieras casi nulas y duraciones muy cortas (promediando 8.6 minutos en el clúster 6). Esto indica que se trata principalmente de cortometrajes, proyectos independientes estudiantiles o registros con datos incompletos en la plataforma original.

Estos descubrimientos son altamente estratégicos para CineVision Studios, especialmente al evaluar el riesgo frente al retorno de inversión (ROI). Aunque los mega-blockbusters del Clúster 5 generan los mayores ingresos brutos, exigen arriesgar capitales corporativos masivos. La principal sugerencia para el estudio es centrar su modelo de negocio en las producciones del Clúster 3: películas con inversiones más seguras y controladas que, al mantener una buena calidad, aseguran rentabilidades altas y constantes sin exponer a la compañía al riesgo financiero de una mega-producción fallida

2. Reglas de asociación

```
# Seleccionar variables y limpiar
datos_reglas <- movies[, c("popularity", "budget", "revenue", "runtime", "voteCount", "voteAvg")]
datos_reglas <- datos_reglas[complete.cases(datos_reglas),]

# Discretizar (Convertir números a texto: Bajo, Medio, Alto)
set.seed(123)
```

```r
datos_disc <- datos_reglas %>%
  mutate(across(everything(), ~ discretize(.x, method = "cluster", breaks = 3, labels = c("Bajo", "Medi

# Convertir a transacciones
trans <- as(datos_disc, "transactions")

# Gráfica para ver frecuencias
itemFrequencyPlot(trans, topN=10, cex.names=0.8, main="Top 10 Categorías más frecuentes", col="steelblu
```




```r
# Generar reglas con dos niveles distintos
# Prueba A: Muy seguras pero menos comunes
reglas_A <- apriori(trans, parameter = list(support = 0.05, confidence = 0.90, target = "rules", minlen=
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.9    0.1    1 none FALSE            TRUE       5    0.05      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: 994
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[18 item(s), 19883 transaction(s)] done [0.00s].
## sorting and recoding items ... [13 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [307 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
reglas_A_ordenadas <- sort(reglas_A, by = "lift")

# Prueba B: Más comunes pero un poco menos seguras
reglas_B <- apriori(trans, parameter = list(support = 0.15, confidence = 0.70, target = "rules", minlen=
```

```
## Apriori
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.7    0.1    1 none FALSE            TRUE       5    0.15      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: 2982
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[18 item(s), 19883 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 5 6 done [0.00s].
## writing ... [301 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
reglas_B_ordenadas <- sort(reglas_B, by = "lift")

# Imprimir los resultados en consola para interpretarlos
cat("\n--- TOP 5 REGLAS A (Soporte 5%, Confianza 90%) ---\n")
```

```
##
## --- TOP 5 REGLAS A (Soporte 5%, Confianza 90%) ---
```

```
inspect(head(reglas_A_ordenadas, 5))
```

```
##     lhs                   rhs             support confidence  coverage     lift count
## [1] {runtime=Bajo,
##      voteAvg=Bajo}     => {budget=Bajo} 0.3251521          1 0.3251521 1.133516  6465
## [2] {runtime=Bajo,
##      voteCount=Bajo,
```

```
##      voteAvg=Bajo}     => {budget=Bajo} 0.3251521          1 0.3251521 1.133516  6465
## [3] {revenue=Bajo,
##      runtime=Bajo,
##      voteAvg=Bajo}     => {budget=Bajo} 0.3251521          1 0.3251521 1.133516  6465
## [4] {popularity=Bajo,
##      runtime=Bajo,
##      voteAvg=Bajo}     => {budget=Bajo} 0.3251521          1 0.3251521 1.133516  6465
## [5] {revenue=Bajo,
##      runtime=Bajo,
##      voteCount=Bajo,
##      voteAvg=Bajo}     => {budget=Bajo} 0.3251521          1 0.3251521 1.133516  6465
```

```
cat("\n--- TOP 5 REGLAS B (Soporte 15%, Confianza 70%) ---\n")
```

```
##
## --- TOP 5 REGLAS B (Soporte 15%, Confianza 70%) ---
```

```
inspect(head(reglas_B_ordenadas, 5))
```

```
##     lhs                   rhs              support confidence  coverage     lift count
## [1] {popularity=Bajo,
##      budget=Bajo,
##      runtime=Bajo}     => {voteAvg=Bajo} 0.3251521  0.8370016 0.3884726 1.976497  6465
## [2] {popularity=Bajo,
##      budget=Bajo,
##      runtime=Bajo,
##      voteCount=Bajo}   => {voteAvg=Bajo} 0.3251521  0.8370016 0.3884726 1.976497  6465
## [3] {popularity=Bajo,
##      budget=Bajo,
##      revenue=Bajo,
##      runtime=Bajo}     => {voteAvg=Bajo} 0.3251521  0.8370016 0.3884726 1.976497  6465
## [4] {popularity=Bajo,
##      budget=Bajo,
##      revenue=Bajo,
##      runtime=Bajo,
##      voteCount=Bajo}   => {voteAvg=Bajo} 0.3251521  0.8370016 0.3884726 1.976497  6465
## [5] {popularity=Bajo,
##      revenue=Bajo,
##      runtime=Bajo}     => {voteAvg=Bajo} 0.3251521  0.8366766 0.3886234 1.975729  6465
```

Eliminación de variables frecuentes

```
# Quitar popularity y voteCount porque sesgan el modelo hacia lo "Bajo"
datos_limpios <- dplyr::select(datos_disc, -popularity, -voteCount)
trans_limpias <- as(datos_limpios, "transactions")

# Generar nuevas reglas buscando relaciones entre presupuesto, duración y éxito
reglas_limpias <- apriori(trans_limpias, parameter = list(support = 0.05, confidence = 0.60, target = "
```

```
## Apriori
```

```
##
## Parameter specification:
##  confidence minval smax arem  aval originalSupport maxtime support minlen
##         0.6    0.1    1 none FALSE            TRUE       5    0.05      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: 994
##
## set item appearances ...[0 item(s)] done [0.00s].
## set transactions ...[12 item(s), 19883 transaction(s)] done [0.00s].
## sorting and recoding items ... [10 item(s)] done [0.00s].
## creating transaction tree ... done [0.00s].
## checking subsets of size 1 2 3 4 done [0.00s].
## writing ... [58 rule(s)] done [0.00s].
## creating S4 object  ... done [0.00s].
```

```
reglas_limpias_ordenadas <- sort(reglas_limpias, by = "lift")

cat("\n--- NUEVAS REGLAS (Sin Popularity ni VoteCount) ---\n")
```

```
##
## --- NUEVAS REGLAS (Sin Popularity ni VoteCount) ---
```

```
inspect(head(reglas_limpias_ordenadas, 5))
```

```
##     lhs                rhs                support confidence   coverage     lift count
## [1] {budget=Medio}  => {voteAvg=Medio} 0.06573455  0.6981838 0.09415078 2.032204  1307
## [2] {budget=Bajo,
##      runtime=Bajo}  => {voteAvg=Bajo}  0.32515214  0.8355952 0.38912639 1.973176  6465
## [3] {budget=Bajo,
##      revenue=Bajo,
##      runtime=Bajo}  => {voteAvg=Bajo}  0.32515214  0.8355952 0.38912639 1.973176  6465
## [4] {revenue=Bajo,
##      runtime=Bajo}  => {voteAvg=Bajo}  0.32515214  0.8352713 0.38927727 1.972411  6465
## [5] {budget=Bajo,
##      voteAvg=Bajo}  => {runtime=Bajo}  0.32515214  0.7679971 0.42337675 1.972115  6465
```

La ejecución inicial del algoritmo Apriori reveló que las asociaciones estaban monopolizadas por atributos en su nivel "Bajo". Al evaluar las frecuencias, se evidenció que categorías como popularity=Bajo y voteCount=Bajo dominaban casi el 95% de las transacciones, reflejando un conjunto de datos saturado de producciones sin impacto comercial. Atendiendo a la necesidad de obtener insights estratégicos, se tomó la decisión metodológica de eliminar ambas variables del modelo. Al suprimir este ruido estadístico, el algoritmo logró descubrir correlaciones subyacentes mucho más valiosas sobre la dinámica de inversión y éxito. Por ejemplo, en las nuevas reglas generadas se descubrió que una película con un presupuesto Medio (budget=Medio) deriva sistemáticamente en una calificación crítica promedio o aceptable (voteAvg=Medio), hallazgo respaldado por una confianza del 69.8% y un indicador Lift de 2.03. Esta depuración confirma que las reglas de asociación son más efectivas cuando se aísla el ruido de las producciones irrelevantes, permitiendo a CineVision Studios identificar los verdaderos impulsores del éxito o fracaso en taquilla.

Análisis de Componentes Principales (PCA)

3.1 Estudie si es posible hacer transformaciones en las variables categóricas para incluirlas en el PCA, ¿valdrá la pena?

Técnicamente es posible transformar variables categóricas (como el idioma original, géneros o casa productora) utilizando métodos como One-Hot Encoding o variables dummy para forzar su inclusión. Sin embargo, no vale la pena ni es metodológicamente correcto para un PCA estándar. El Análisis de Componentes Principales está diseñado para maximizar la varianza lineal en variables continuas basándose en distancias euclidianas. Al introducir docenas o cientos de columnas binarias (ceros y unos), se genera una matriz muy dispersa que distorsiona las correlaciones geométricas, diluye el peso de las variables financieras verdaderamente importantes e imposibilita una interpretación clara de los componentes.

3.2 Estudie si es conveniente hacer un Análisis de Componentes Principales. Recuerde que puede usar el índice KMO y el test de esfericidad de Bartlett.

```
# Calcular la matriz de correlación
rcor <- cor(datos_scaled)

# Calcular el determinante de la matriz
print(paste("Determinante de la matriz:", det(rcor)))
```

```
## [1] "Determinante de la matriz: 0.0774134562712613"
```

```
# Test de Esfericidad de Bartlett
test_bartlett <- cortest.bartlett(rcor, n = nrow(datos_scaled))
print("Test de Bartlett:")
```

```
## [1] "Test de Bartlett:"
```

```
print(test_bartlett)
```

```
## $chisq
## [1] 50862.73
##
## $p.value
## [1] 0
##
## $df
## [1] 15
```

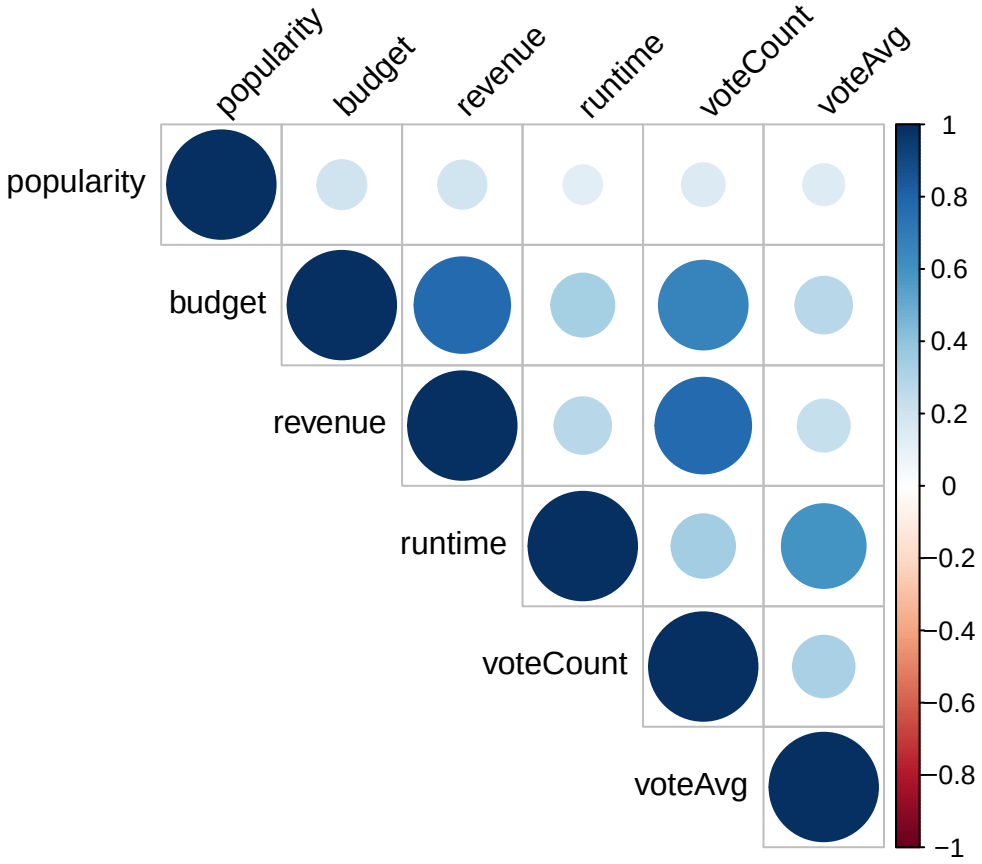
```
# Índice KMO
indice_kmo <- KMO(rcor)
print("Índice KMO:")
```

```
## [1] "Índice KMO:"
```

```r
print(indice_kmo)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = rcor)
## Overall MSA =  0.73
## MSA for each item =
## popularity     budget    revenue    runtime  voteCount    voteAvg
##       0.90       0.79       0.69       0.68       0.79       0.66
```

```r
# Gráfico de la matriz de correlación
corrplot(rcor, method = "circle", type = "upper", tl.col = "black", tl.srt = 45)
```

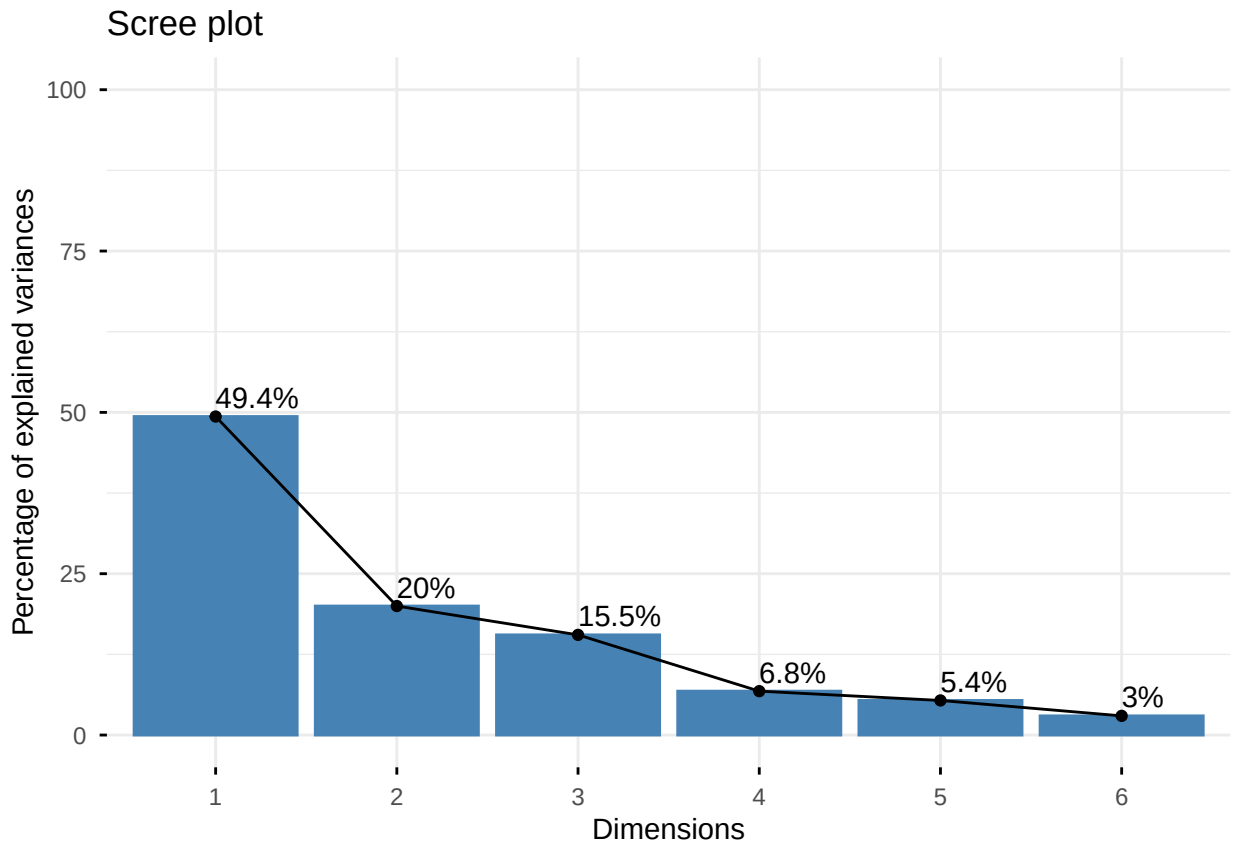


Para determinar la viabilidad y conveniencia de aplicar el Análisis de Componentes Principales (PCA), se evaluó la matriz de correlación de las variables numéricas mediante tres métricas estadísticas clave. Primero, el determinante de la matriz arrojó un valor de 0.0774, una cifra cercana a cero que indica la existencia de multicolinealidad, evidenciando que las variables están relacionadas linealmente entre sí. Seguidamente, el test de esfericidad de Bartlett confirmó esta relación al obtener un valor p de 0 (con un estadístico de 50862.73), lo que permite rechazar de forma contundente la hipótesis nula de que nos encontramos ante una matriz identidad, demostrando que existe una correlación suficientemente fuerte para realizar el análisis. Finalmente, el índice Kaiser-Meyer-Olkin (KMO) resultó en una adecuación muestral general (Overall MSA) de 0.73 (con valores individuales que alcanzan hasta el 0.90 en popularidad), un valor clasificado como "bueno" y muy superior al umbral de aceptación de 0.5. En conclusión, los tres indicadores respaldan sólidamente que el conjunto de datos posee la estructura relacional idónea y que es altamente conveniente proceder con la reducción de dimensionalidad mediante el PCA.

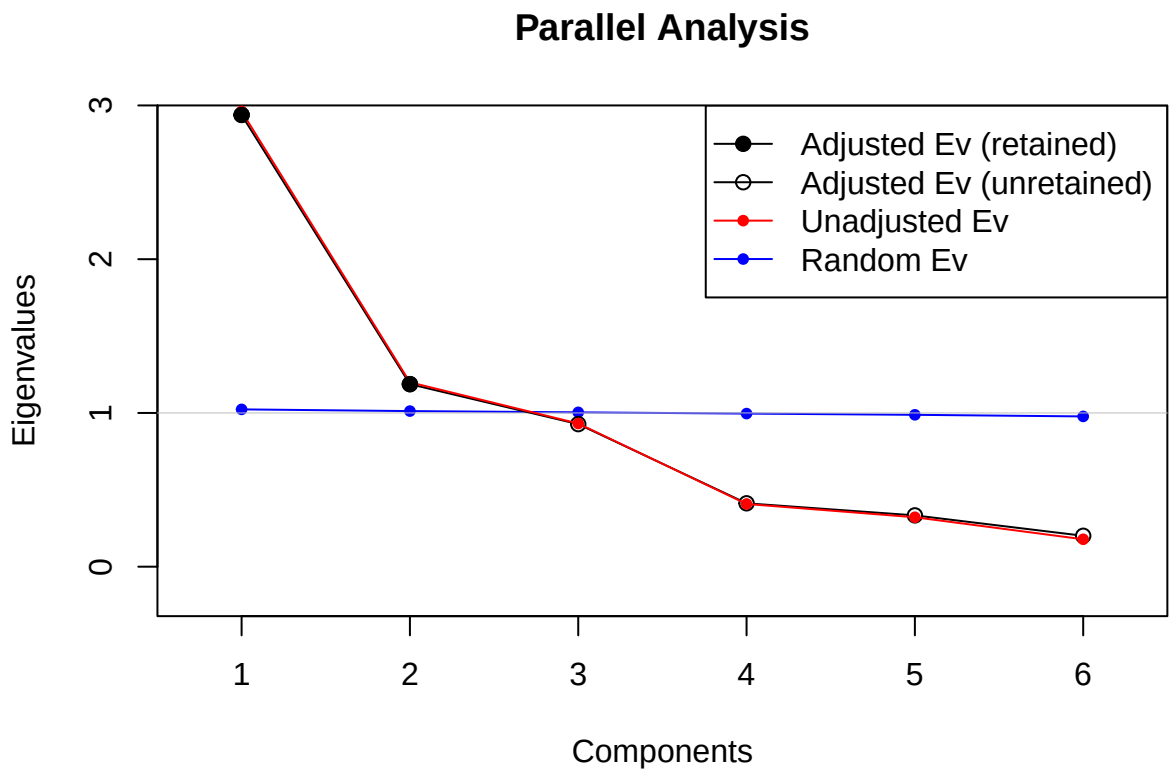
3.3 Haga un análisis de componentes principales con las variables numéricas, discuta los resultados e interprete los componentes.

```
# Aplicar PCA
compPrinc <- prcomp(datos_scaled, scale = FALSE)

# Resumen de varianza explicada
summary(compPrinc)

# Valores propios (Regla de Kaiser)
valores_propios <- compPrinc$sdev^2
print("Valores propios:")
print(valores_propios)

# Gráfico de sedimentación (Scree Plot)
fviz_eig(compPrinc, addlabels = TRUE, ylim = c(0, 100))
```

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

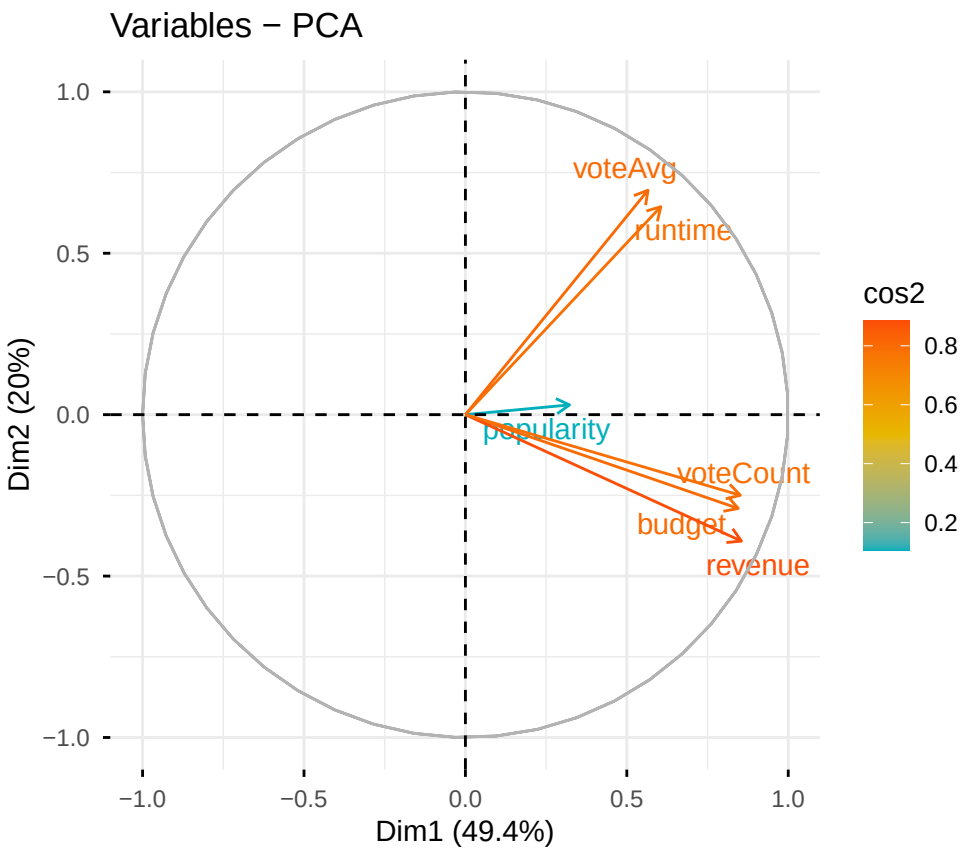



```
# Prueba de Paralelismo (Horn's Parallel Analysis)
paran(datos_scaled, graph=TRUE)
```

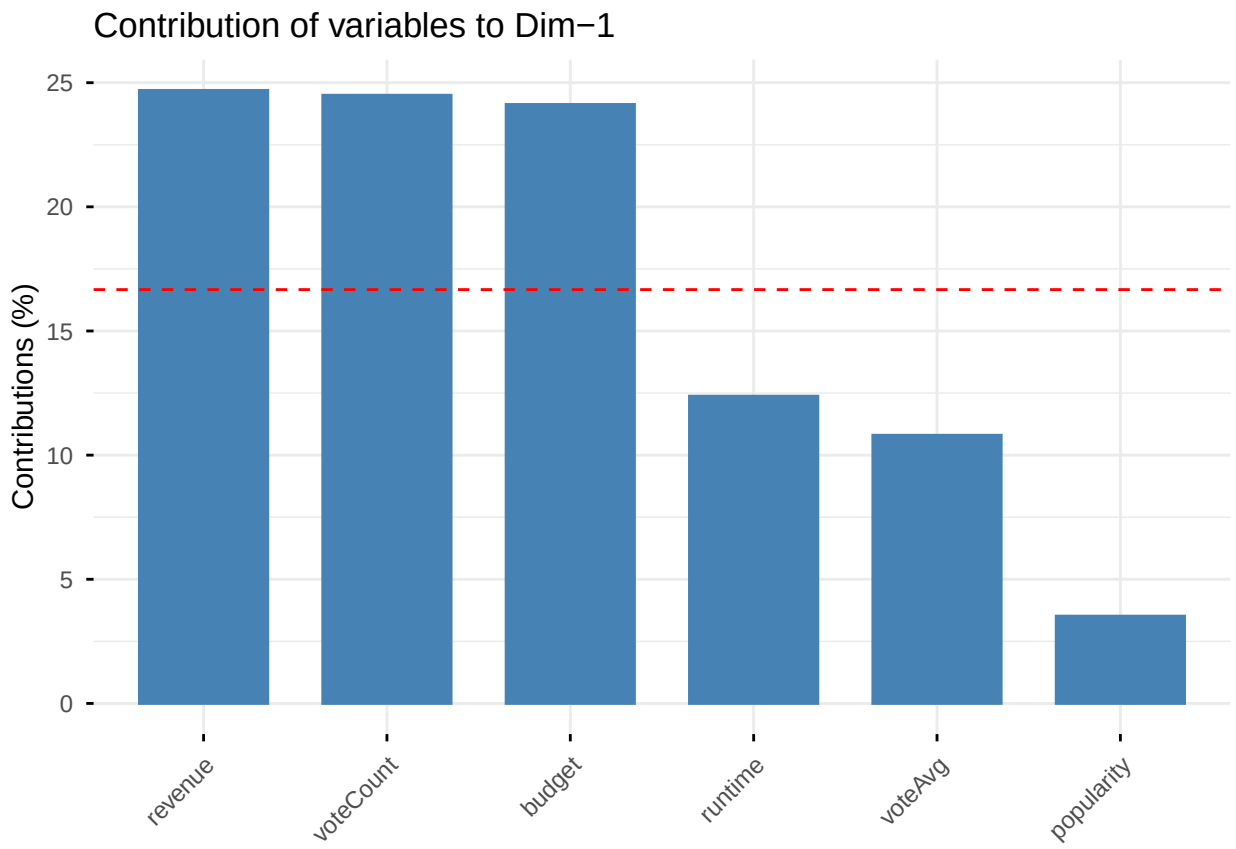



```
# --- Interpretación Visual ---
# Gráfico de calidad de representación (Coseno cuadrado)
fviz_pca_var(compPrinc, col.var = "cos2",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE)
```

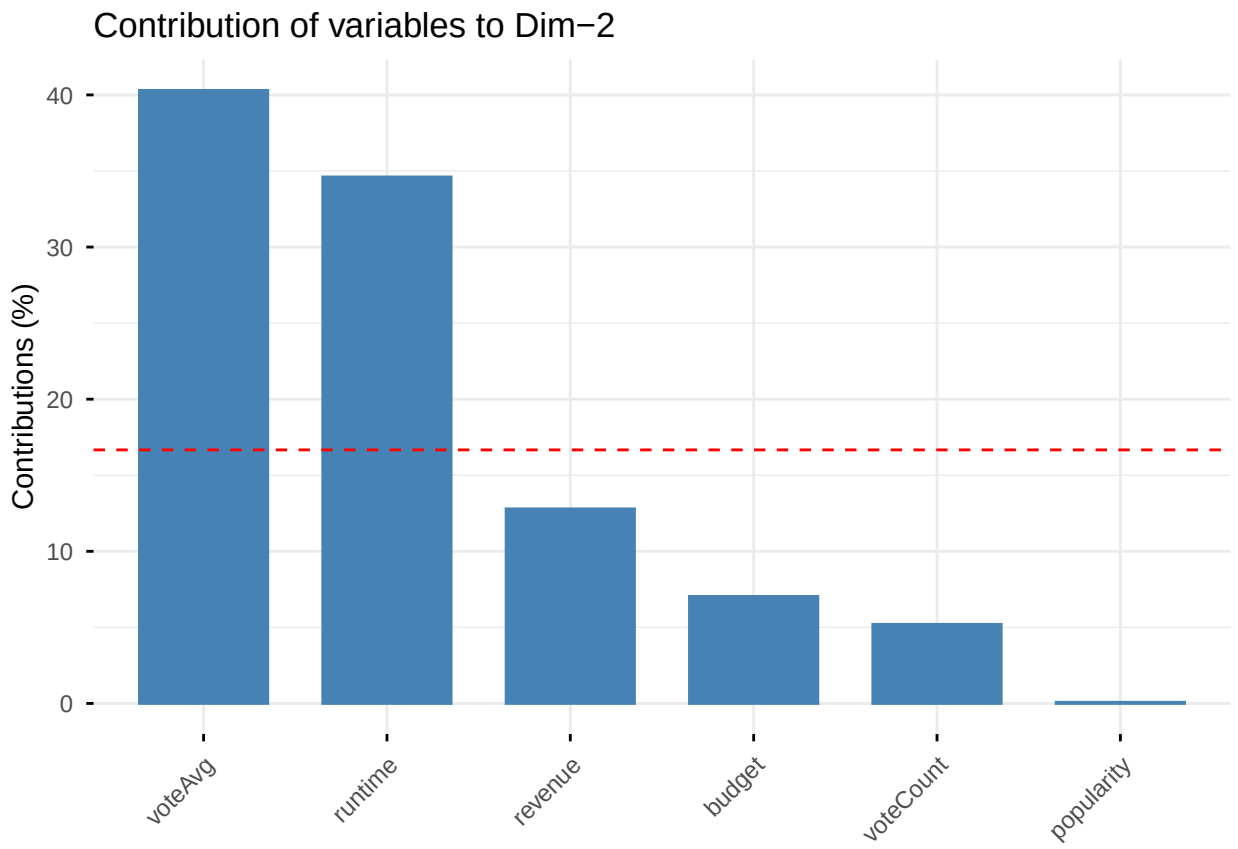
```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## i The deprecated feature was likely used in the ggpubr package.
##   Please report the issue at <https://github.com/kassambara/ggpubr/issues>.
## This warning is displayed once per session.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```r
# Contribución de las variables a las primeras 2 dimensiones
fviz_contrib(compPrinc, choice = "var", axes = 1, top = 6)
```




```
fviz_contrib(compPrinc, choice = "var", axes = 2, top = 6)
```



Para determinar la cantidad de componentes a retener, se evaluó la Regla de Kaiser y la Prueba de Paralelismo de Horn. Ambas métricas indicaron conservar únicamente las dos primeras dimensiones (con valores propios de 2.96 y 1.19, respectivamente). Al seleccionar estos dos componentes, se logra explicar de manera acumulada el 69.35% de la varianza total de los datos, logrando una reducción de dimensionalidad sumamente eficiente con muy poca pérdida de información. Al analizar el círculo de correlaciones y la contribución de las variables, la naturaleza de las dimensiones queda clara. El Primer Componente (PC1) explica el 49.36% de la varianza y se define como la "Magnitud Comercial", ya que está fuertemente impulsado por revenue, voteCount y budget, resumiendo el músculo económico y de taquilla de la cinta. Por su parte, el Segundo Componente (PC2) explica el 19.99% restante y representa la "Recepción Crítica", al estar dominado por la calificación (voteAvg) y la duración (runtime). Esto demuestra que la calidad percibida de una película es independiente de los millones invertidos en ella. Finalmente, la variable popularity mostró poca representación en estos ejes, sugiriendo que la viralidad es un factor independiente y volátil.