Clusterização usando K-Médias, K-Medóides e Agrupamento Hierárquico

Autora: Maria Francinete Mateus

OBJETIVO:

Agrupar 100 filmes produzidos (total ou parcialmente) pelo Reino Unido, para encontrar insights que possam ser usados no processo de seleção e criação de conteúdo; neste caso, para compor um guia de viagens inspiradas no cinema.

DADOS:

Dados públicos disponibilizados pelo IMDb no site do Kaggle. Um dos problemas desse tipo de base de dados é que cada filme pode ter até 3 tipos de gêneros diferentes, o que dificulta a separação e classificação única deles usando ferramentas mais simples (por exemplo, o Excel).

PARTE 1: CARREGAMENTO DE PACOTES & DATASET

```
library(readxl)
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(ggplot2)
library(dummies)
```

```
## dummies-1.5.6 provided by Decision Patterns
```

```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.6     v purrr   0.3.4
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

```
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(cluster)
library(factoextra)
```

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

```
library(fpc)
```

```
Base_para_clustering_filmes <- read_excel("Base-para-clustering-filmes.xlsx")
```

PARTE 2: EXPLORAÇÃO E PADRONIZAÇÃO DOS DADOS

```
str(Base_para_clustering_filmes)
```

```
## tibble [100 x 11] (S3: tbl_df/tbl/data.frame)
##  $ Filmes         : chr [1:100] "F1" "F2" "F3" "F4" ...
##  $ Título original: chr [1:100] "28 Days Later..." "A Monster Calls" "A Street Cat Named B
ob" "About Time" ...
##  $ Anos           : num [1:100] 2002 2016 2016 2013 2009 ...
##  $ Gen1           : chr [1:100] "Action" "Adventure" "Biography" "Comedy" ...
##  $ Gen2           : chr [1:100] "Drama" "Drama" "Drama" "Drama" ...
##  $ Dur_min        : num [1:100] 113 108 103 123 100 129 123 113 100 110 ...
##  $ Notas_md       : num [1:100] 7.6 7.5 7.4 7.8 7.3 7.3 7.8 7.6 7.3 7.7 ...
##  $ Votos          : num [1:100] 369508 78648 26386 289857 126951 ...
##  $ Bilhet_m_$     : num [1:100] 85720 47309 16053 87100 26097 ...
##  $ Rev_usu        : num [1:100] 1521 241 102 651 256 ...
##  $ Rev_crit       : num [1:100] 135 353 77 280 280 255 306 582 148 162 ...
```

```
head(Base_para_clustering_filmes)
```

```
## # A tibble: 6 x 11
##   Filmes `Título origin~`  Anos Gen1  Gen2  Dur_min Notas_md  Votos `Bilhet_m_$`
##   <chr>  <chr>            <dbl> <chr> <chr>   <dbl>    <dbl>  <dbl>        <dbl>
## 1 F1     28 Days Later...  2002 Acti~ Drama     113      7.6 369508       85720.
## 2 F2     A Monster Calls   2016 Adve~ Drama     108      7.5  78648       47309.
## 3 F3     A Street Cat Na~  2016 Biog~ Drama     103      7.4  26386       16053.
## 4 F4     About Time        2013 Come~ Drama     123      7.8 289857       87100.
## 5 F5     An Education      2009 Drama Drama     100      7.3 126951       26097.
## 6 F6     Another Year      2010 Come~ Drama     129      7.3  27487       19723.
## # ... with 2 more variables: Rev_usu <dbl>, Rev_crit <dbl>
```

```
summary(Base_para_clustering_filmes)
```

```
##     Filmes          Título original         Anos          Gen1
##  Length:100         Length:100         Min.   :2000   Length:100
##  Class :character   Class :character   1st Qu.:2006   Class :character
##  Mode  :character   Mode  :character   Median :2010   Mode  :character
##                                        Mean   :2010
##                                        3rd Qu.:2015
##                                        Max.   :2020
##      Gen2              Dur_min         Notas_md         Votos
##  Length:100         Min.   : 85.0   Min.   :7.300   Min.   :  12206
##  Class :character   1st Qu.:105.8   1st Qu.:7.400   1st Qu.:  82107
##  Mode  :character   Median :118.5   Median :7.600   Median : 190687
##                     Mean   :118.8   Mean   :7.645   Mean   : 284840
##                     3rd Qu.:129.0   3rd Qu.:7.800   3rd Qu.: 438695
##                     Max.   :164.0   Max.   :8.500   Max.   :1480582
##    Bilhet_m_$         Rev_usu          Rev_crit
##  Min.   :   8160   Min.   :  18.0   Min.   : 57.0
##  1st Qu.:  34035   1st Qu.: 266.2   1st Qu.:191.5
##  Median :  83012   Median : 594.0   Median :278.0
##  Mean   : 228364   Mean   : 726.8   Mean   :308.6
##  3rd Qu.: 212972   3rd Qu.: 948.2   3rd Qu.:417.2
##  Max.   :1342167   Max.   :3367.0   Max.   :782.0
```

```
any(is.na(Base_para_clustering_filmes))
```

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

```
novoimdb= data.frame(Base_para_clustering_filmes, row.names = 1)
```

```
Gen1dummy = dummy(novoimdb$Gen1)
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

```
Gen2dummy = dummy(novoimdb$Gen2)
```

```
## Warning in model.matrix.default(~x - 1, model.frame(~x - 1), contrasts = FALSE):
## non-list contrasts argument ignored
```

```
Base_combinada = cbind(novoimdb[,c(-1,-2, -3, -4)], Gen1dummy, Gen2dummy)
```

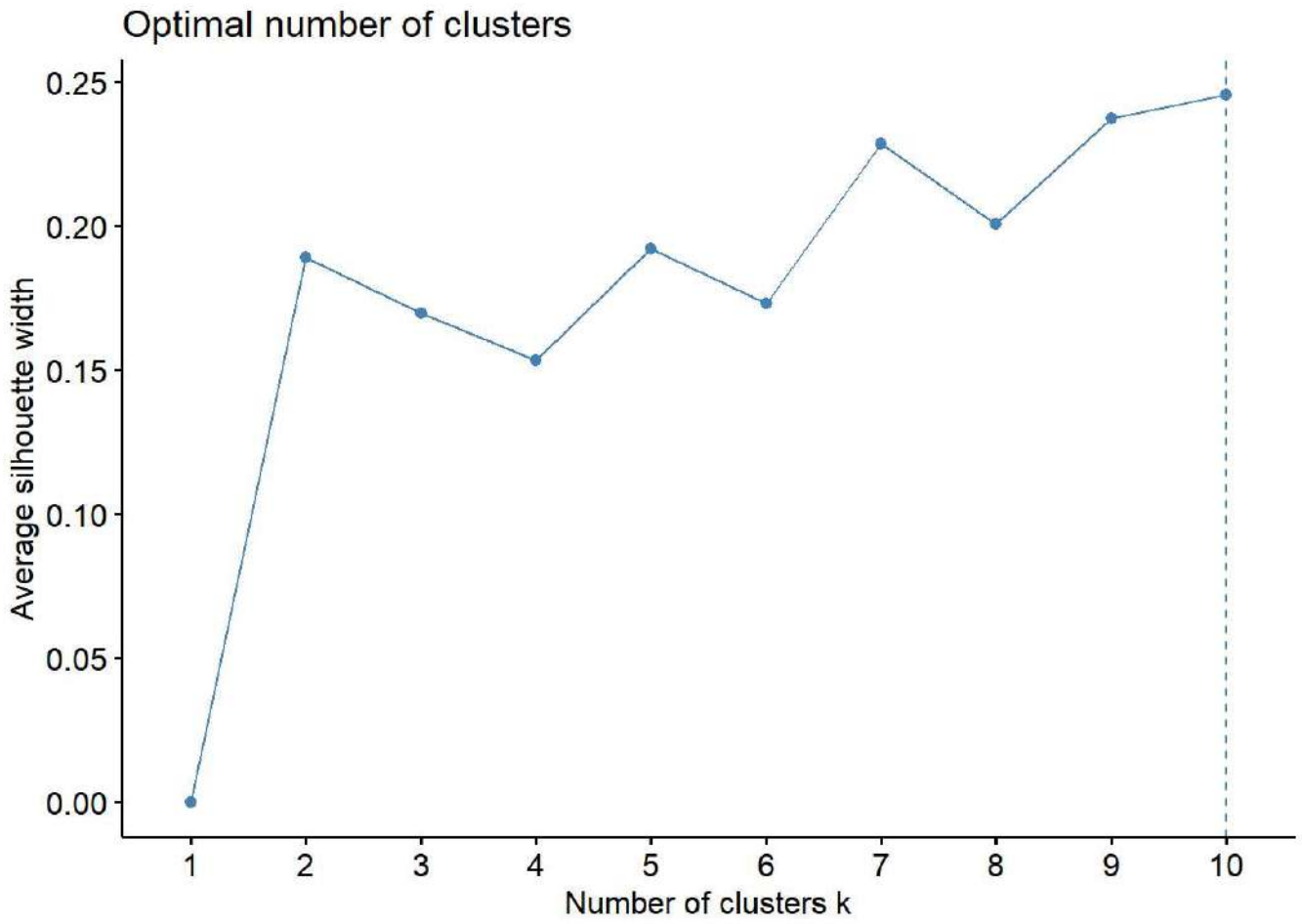
```
dadosnorm = data.frame(scale(Base_combinada))
```

PARTE 3: MODELAGEM DOS DADOS

AGRUPAMENTO COM K-MÉDIAS

Encontrando o nº ideal de Clusters com a métrica Silhouette

```
silhuetemz = fviz_nbclust(dadosnorm, kmeans, method = c("silhouette", "wss", "gap_stat"))
plot(silhuetemz)
```



Número ideal de classes igual a 10

```
silhuetemz$data
```

```
##    clusters         y
## 1         1 0.0000000
## 2         2 0.1890118
## 3         3 0.1697025
## 4         4 0.1533880
## 5         5 0.1920820
## 6         6 0.1729387
## 7         7 0.2286194
## 8         8 0.2005859
## 9         9 0.2374142
## 10       10 0.2453604
```

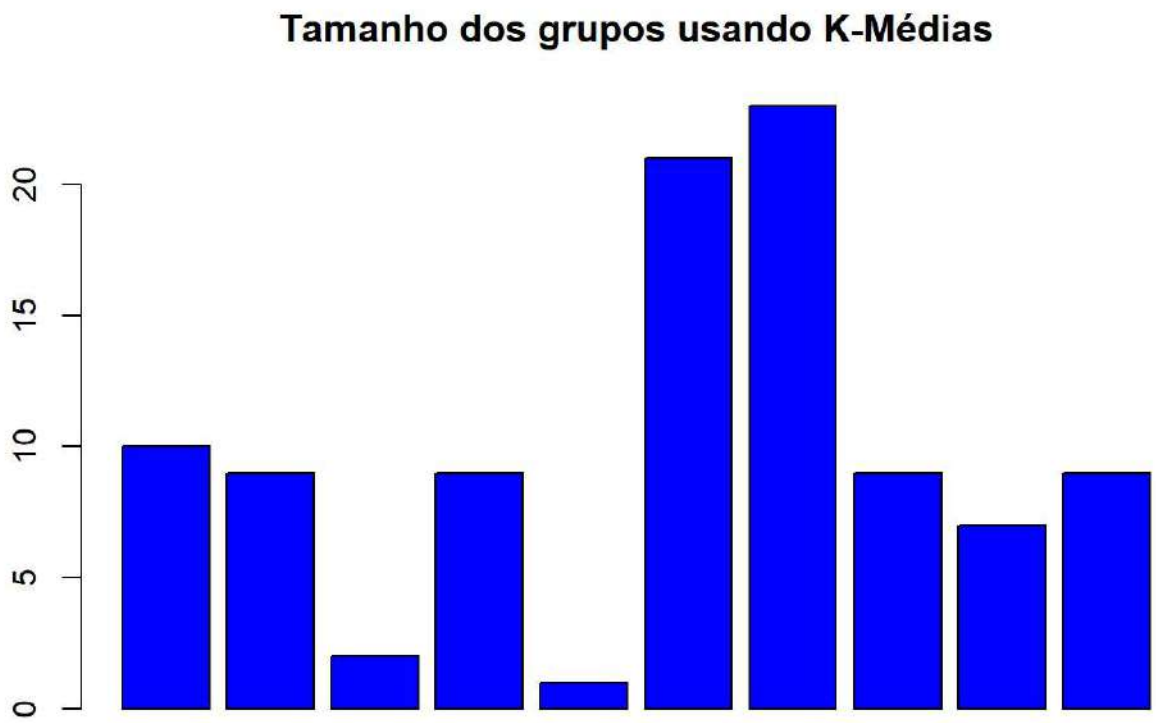
Aplicando o K-Médias com 10 classes, cujo Silhouette é 0.2453604

```
agrkmeans = kmeans(dadosnorm, 10)
```

```
agrkmeans$size
```

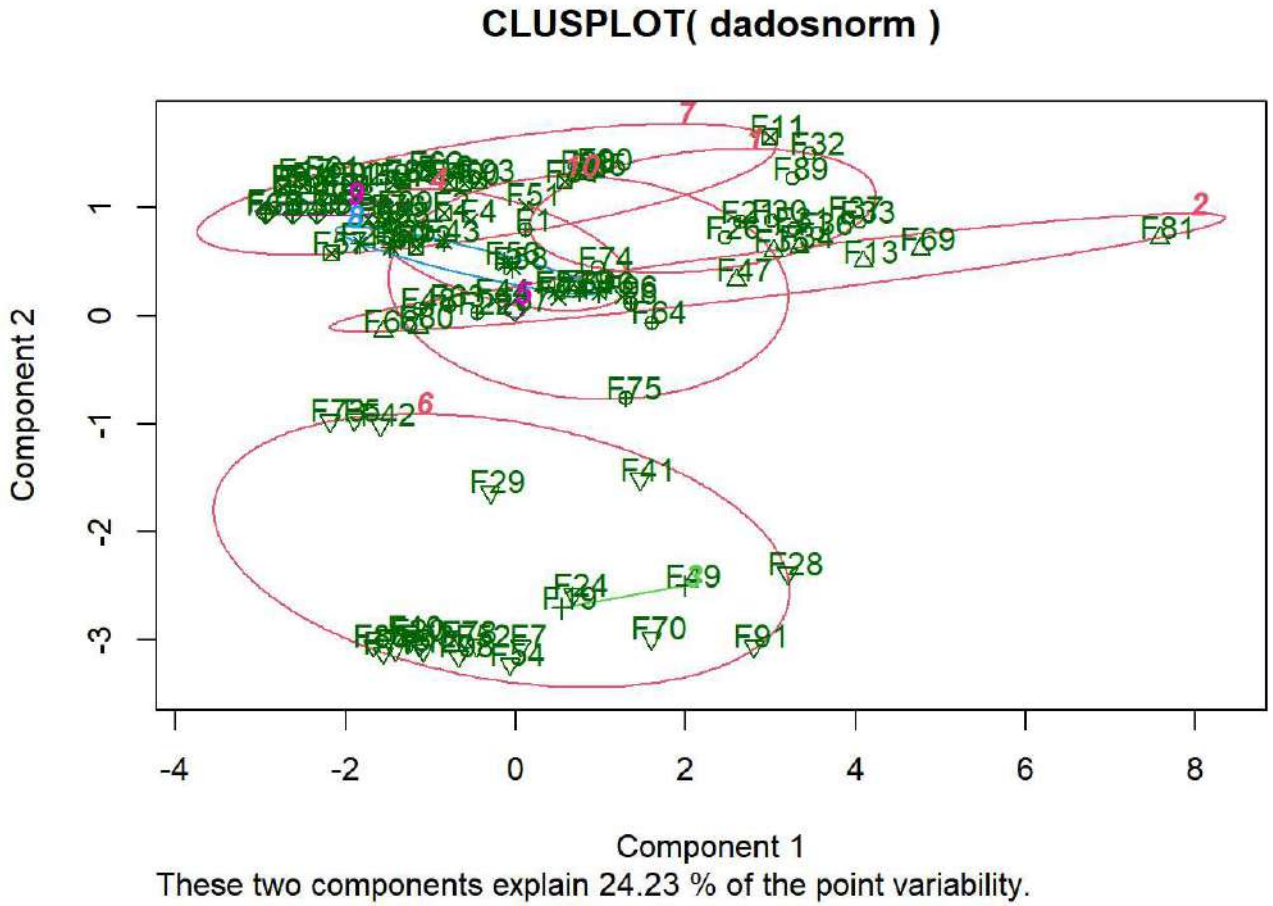
```
##  [1] 10  9  2  9  1 21 23  9  7  9
```

```
barplot(agrkmeans$size, main = "Tamanho dos grupos usando K-Médias", col = "blue")
```

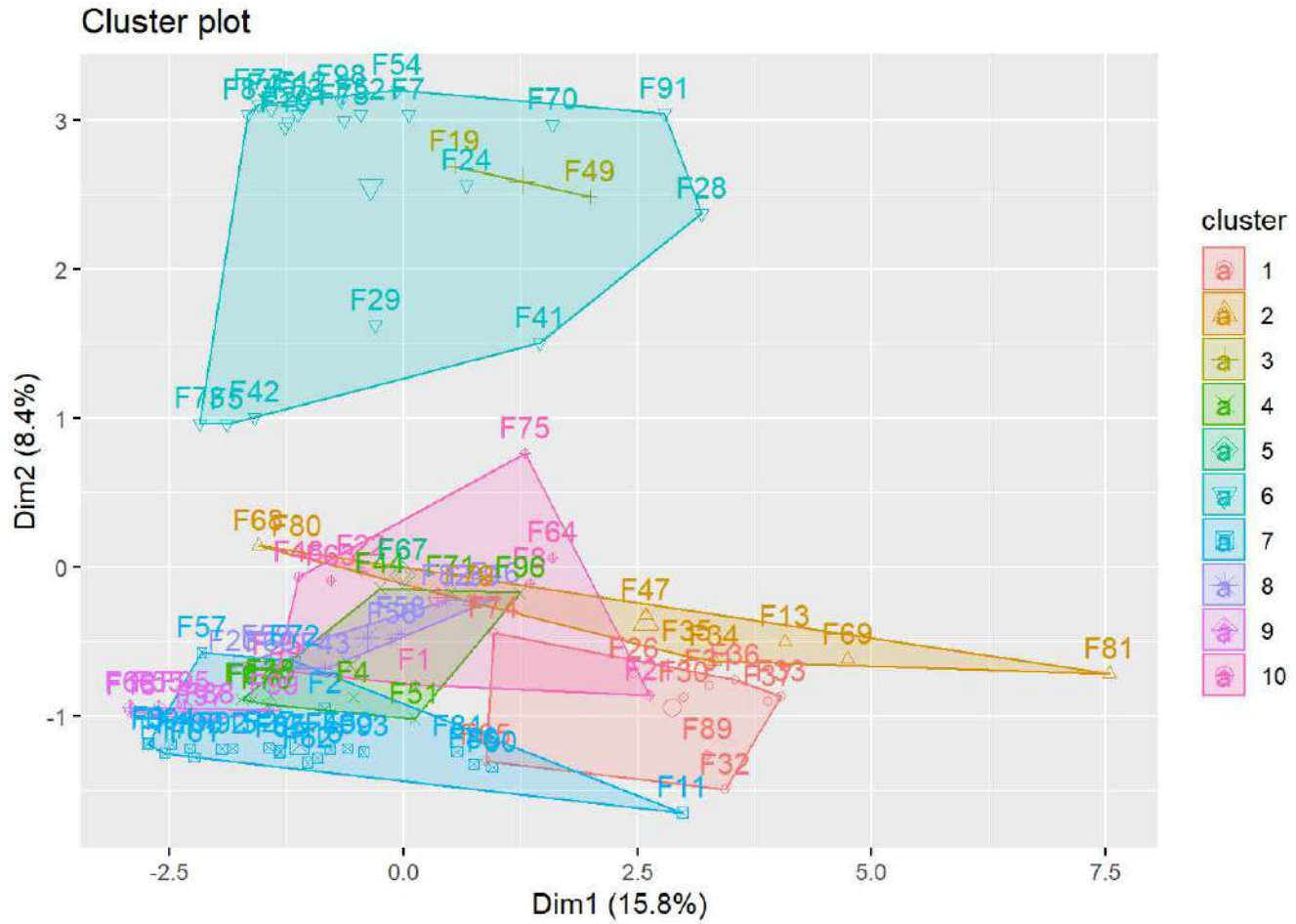


K-Médias apresenta índice Silhouette de 0.245 e distribuição desproporcional dos filmes entre os grupos: 10, 9, 2, 9, 1, 21, 23, 9, 7 e 9.

```
par(mfrow=c(1,1))
clusplot(dadosnorm, agrkmeans$cluster, color = TRUE, labels = 2, lines = 0)
```



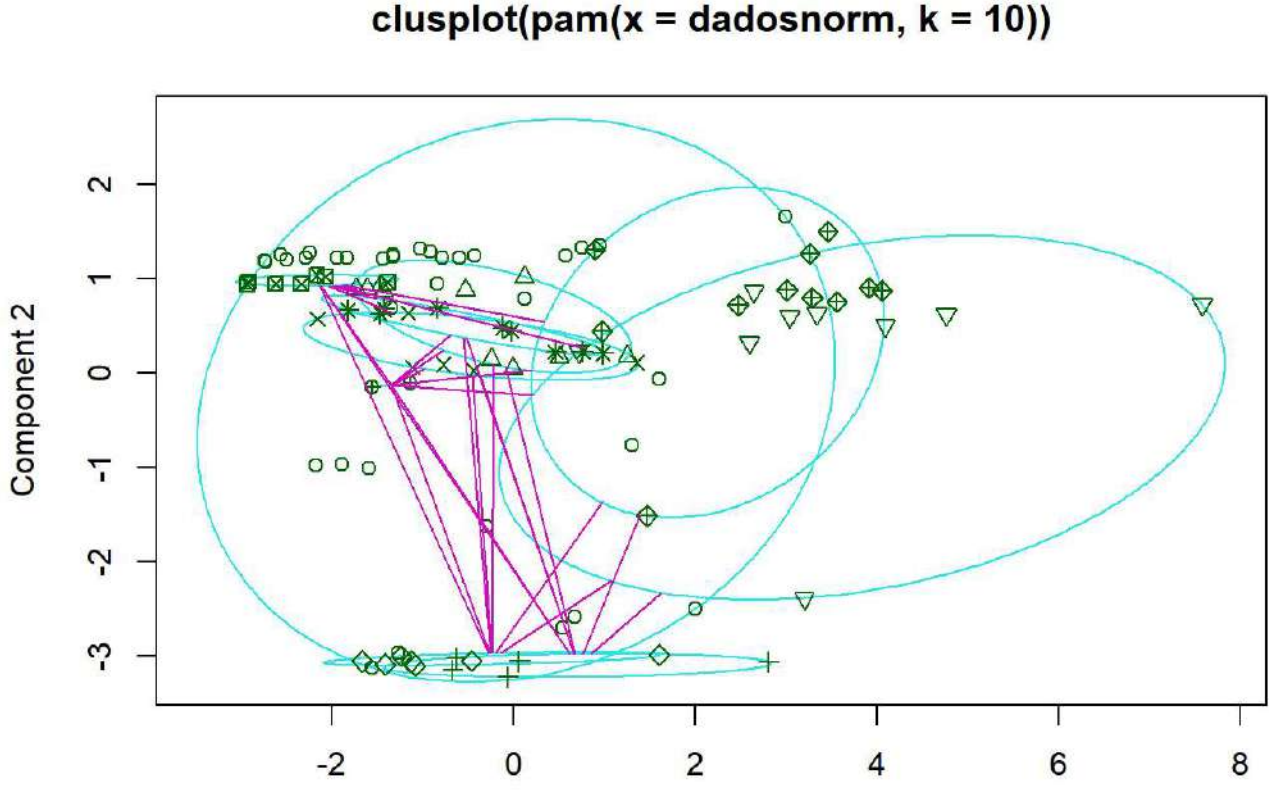

```
fviz_cluster(agrkmeans, data = dadosnorm)
```

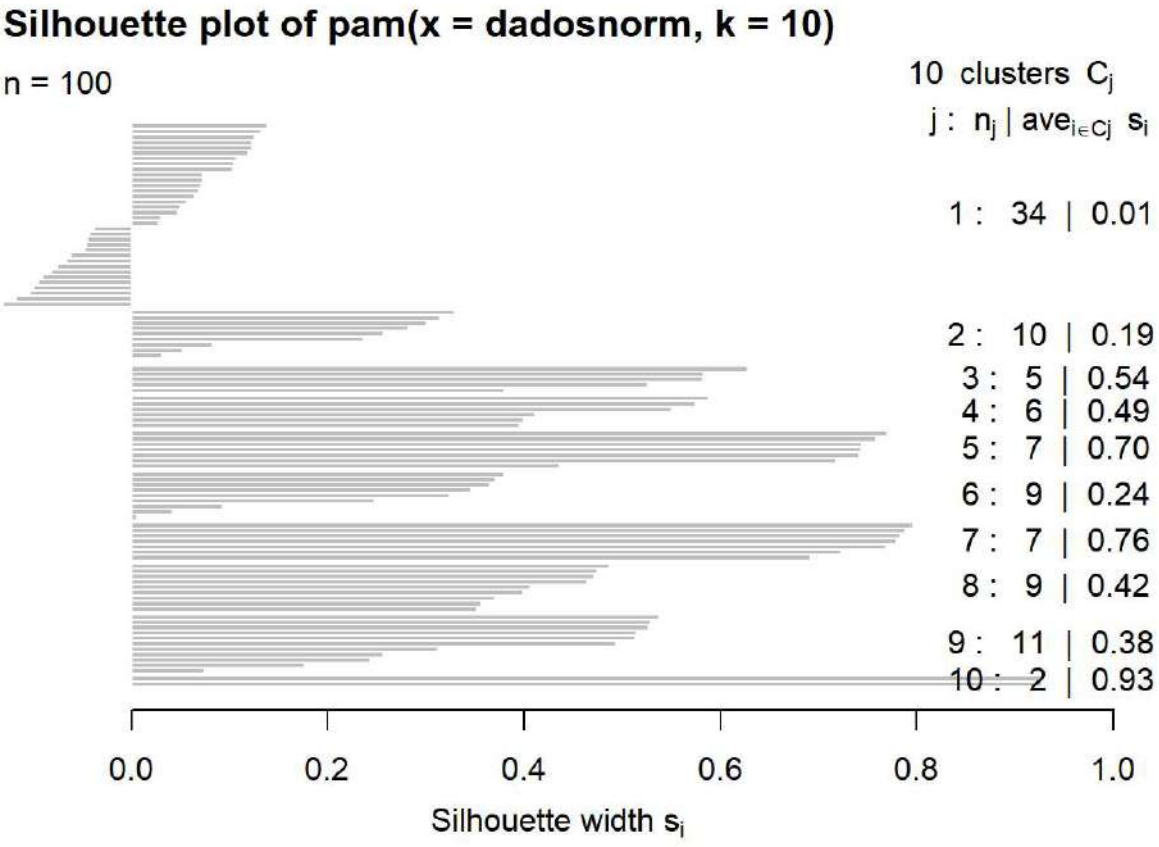


```
Kmedias = agrkmeans$cluster
```

AGRUPAMENTO COM K-MEDÓIDES

```
clusterspam = pam(dadosnorm, 10)
plot(clusterspam)
```


clusplot(pam(x = dadosnorm, k = 10))
Component 2
Component 1
These two components explain 24.23 % of the point variability.


Silhouette plot of pam(x = dadosnorm, k = 10)
n = 100
10 clusters Cj
j : nj | avei∈Cj si
1 : 34 | 0.01
2 : 10 | 0.19
3 : 5 | 0.54
4 : 6 | 0.49
5 : 7 | 0.70
6 : 9 | 0.24
7 : 7 | 0.76
8 : 9 | 0.42
9 : 11 | 0.38
10 : 2 | 0.93
Silhouette width si
Average silhouette width : 0.3

```
par(mfrow=c(2,2))
clusterspam = pam(dadosnorm, 10)
ssp = silhouette(clusterspam$cluster, dist(dadosnorm))
mean(ssp[,3])
```

```
## [1] 0.302332
```

K-Medóides apresenta índice Silhouette de 0.30 e, comparado ao K-Médias, uma distribuição menos desproporcional dos filmes entre os grupos: 34, 10, 5, 6, 7, 9, 7, 9, 11 e 2.

```
clusterspam$clusinfo
```

```
##       size   max_diss   av_diss   diameter separation
##  [1,]   34 11.1481052 4.0337478 14.9470093   3.068956
##  [2,]   10 10.4317882 3.3246854 10.8819150   4.036990
##  [3,]    5  4.4127454 1.8708284  5.6389109   5.158661
##  [4,]    6  3.6869293 2.0798301  5.0961804   4.049660
##  [5,]    7  3.5374477 1.2627865  4.4447045   4.485640
##  [6,]    9  8.7935797 3.6222388  9.8488283   3.991273
##  [7,]    7  2.0241916 1.1154016  2.3322164   4.518575
##  [8,]    9  4.3271812 2.7146579  4.7279381   4.039008
##  [9,]   11  5.0521718 2.9670639  7.2508200   3.068956
## [10,]    2  0.6669028 0.3334514  0.6669028   8.126240
```

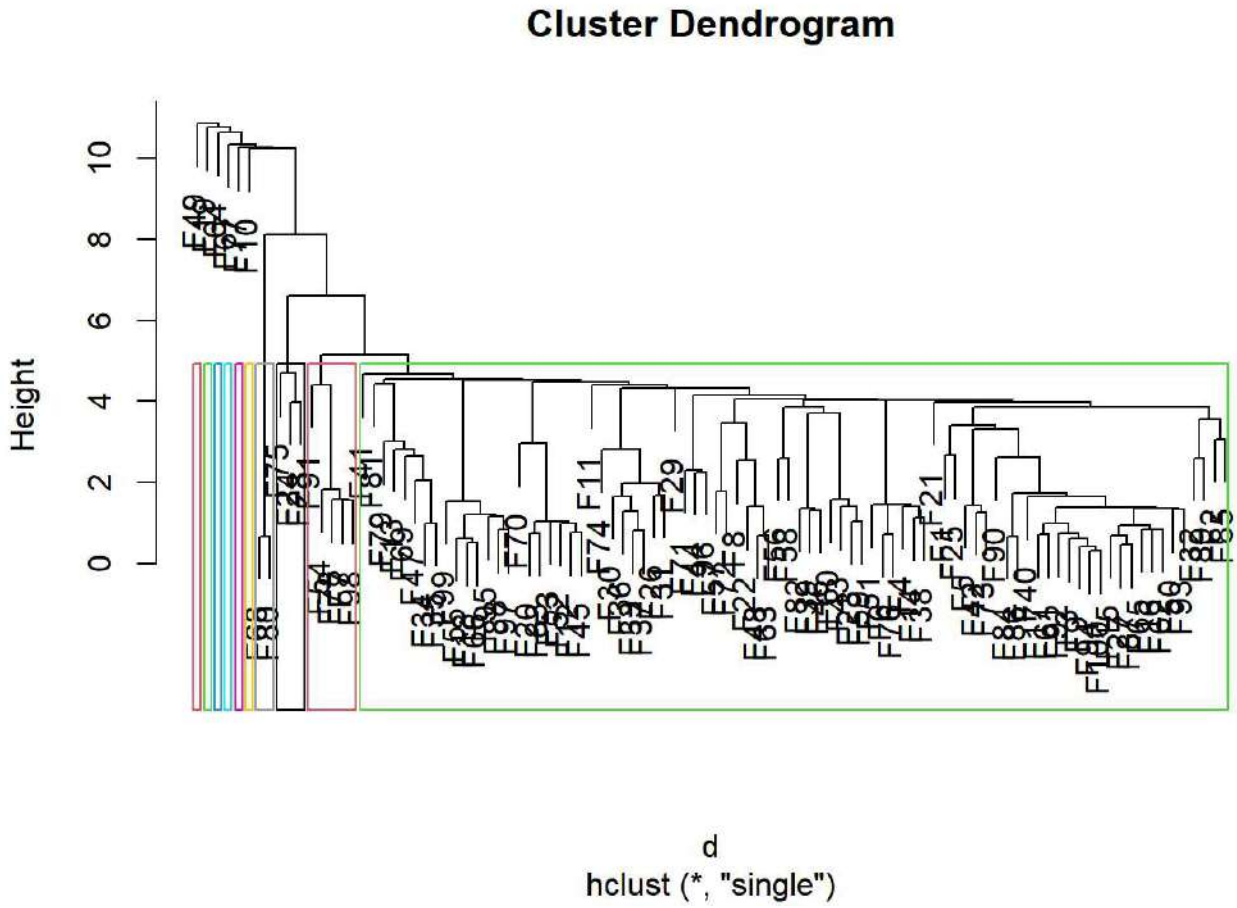
```
kmedoides = clusterspam$clustering
```

AGRUPAMENTOS HIERÁRQUICOS (AH)

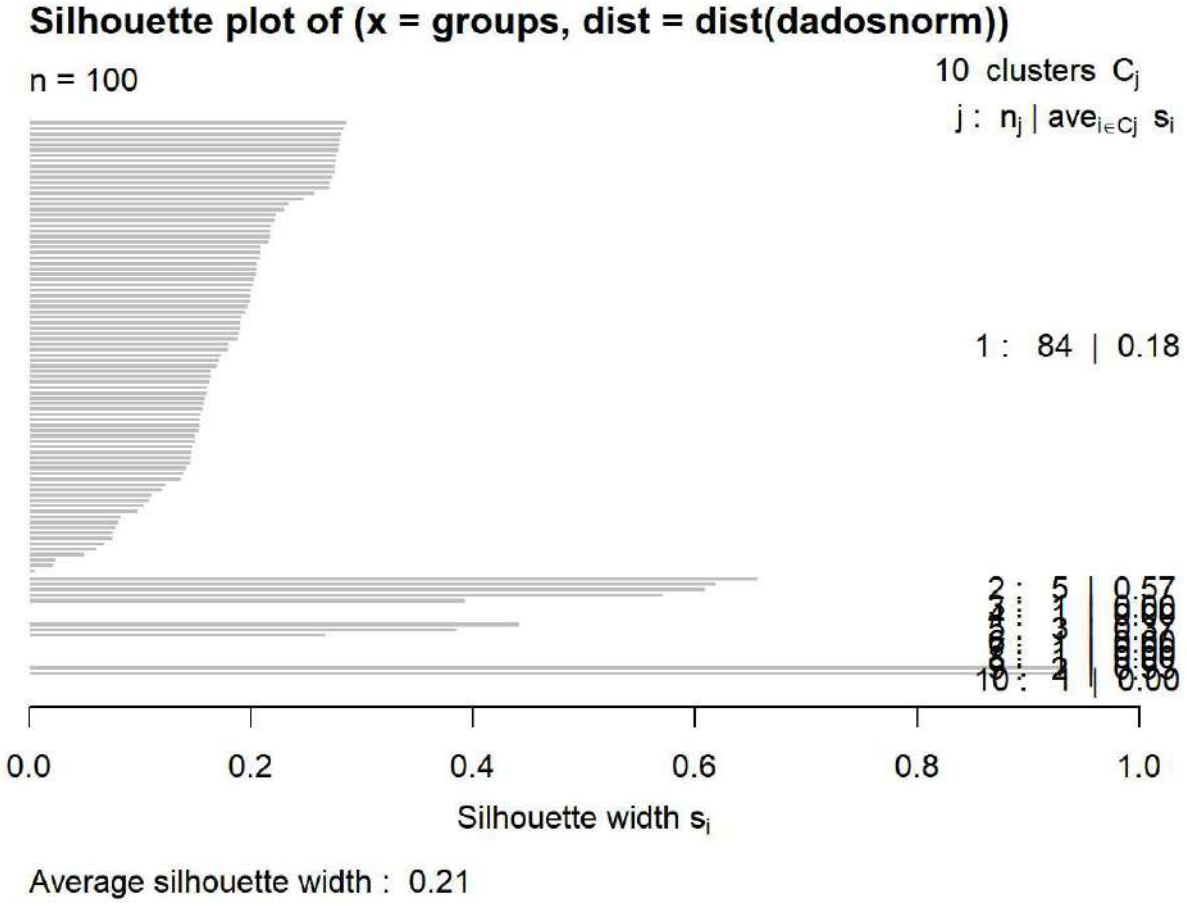
USANDO SINGLE LINKAGE

```
d = dist(dadosnorm, method = "euclidean")
fit = hclust(d, method = "single")

par(mfrow=c(1,1))
plot(fit)
groups = cutree(fit, k=10)
rect.hclust(fit, k=10, border = 2:12)
```




```
sssi = silhouette(groups, dist(dadosnorm))
plot(sssi)
```



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

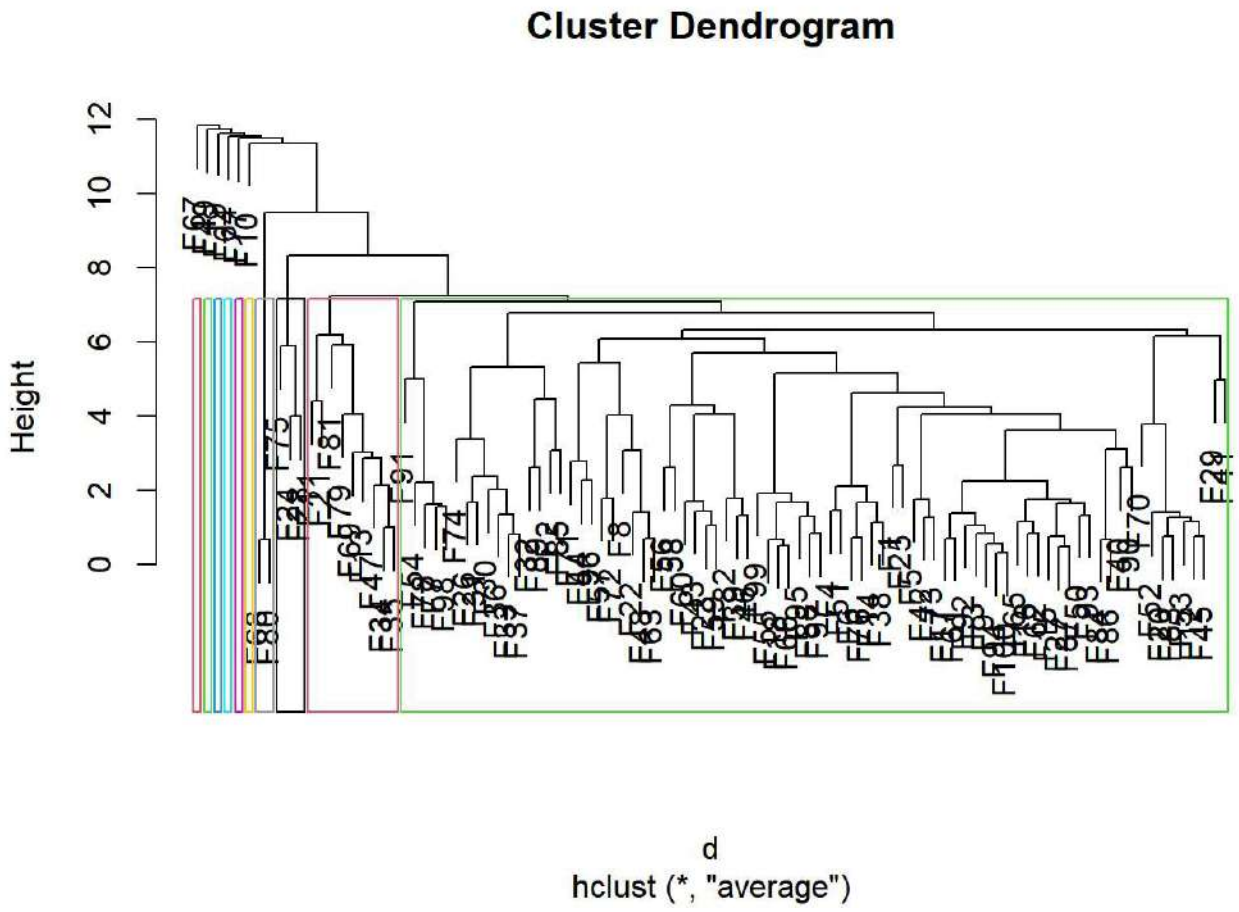
```
## [1] 0.206971
```

Single Linkage apresenta índice Silhouette de 0.21 e grupos muito desproporcionais: 84, 5, 1, 1, 3, 1, 1, 1, 2, 1.

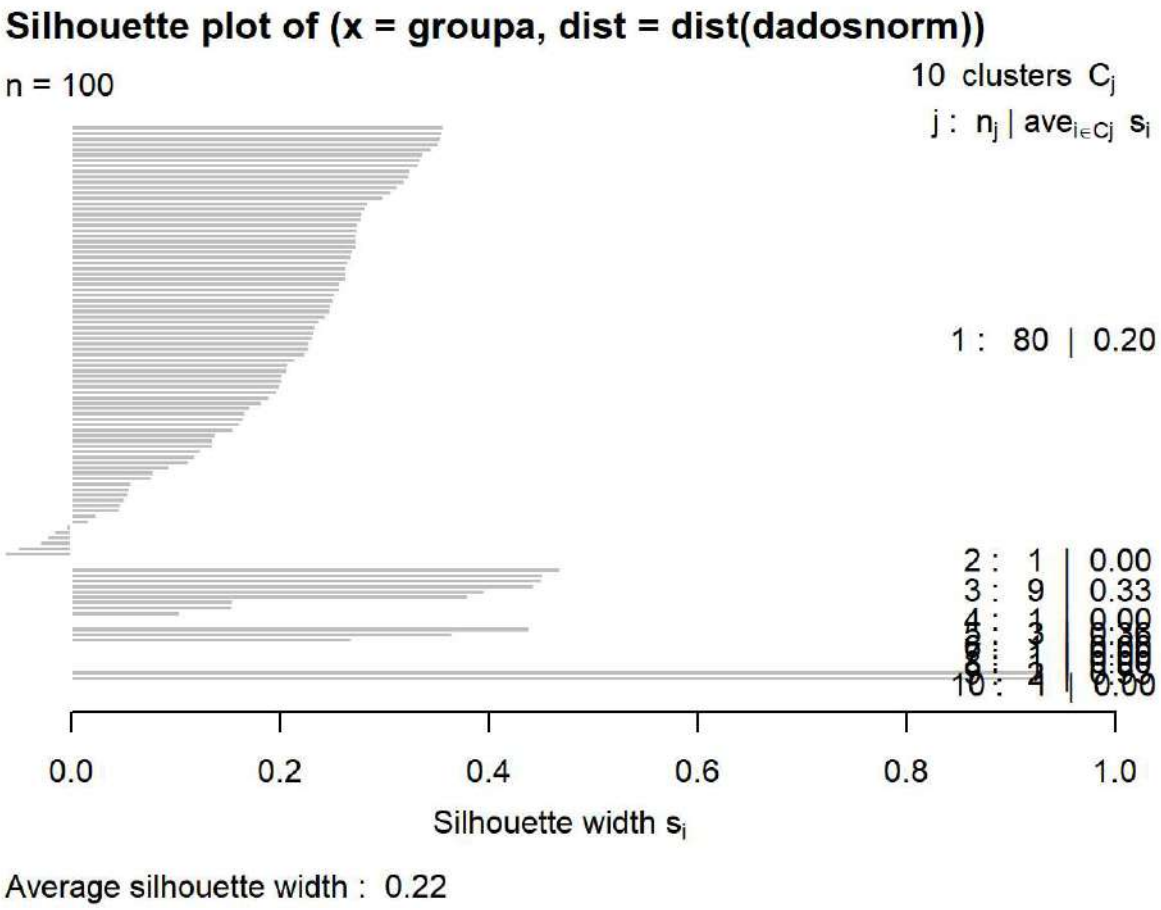
USANDO AVERAGE LINKAGE

```
d = dist(dadosnorm, method = "euclidean")
fita = hclust(d, method = "average")

par(mfrow=c(1,1))
plot(fita)
groupa = cutree(fita, k=10)
rect.hclust(fita, k=10, border= 2:12)
```




```
ssav = silhouette(groupa, dist(dadosnorm))
plot(ssav)
```




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

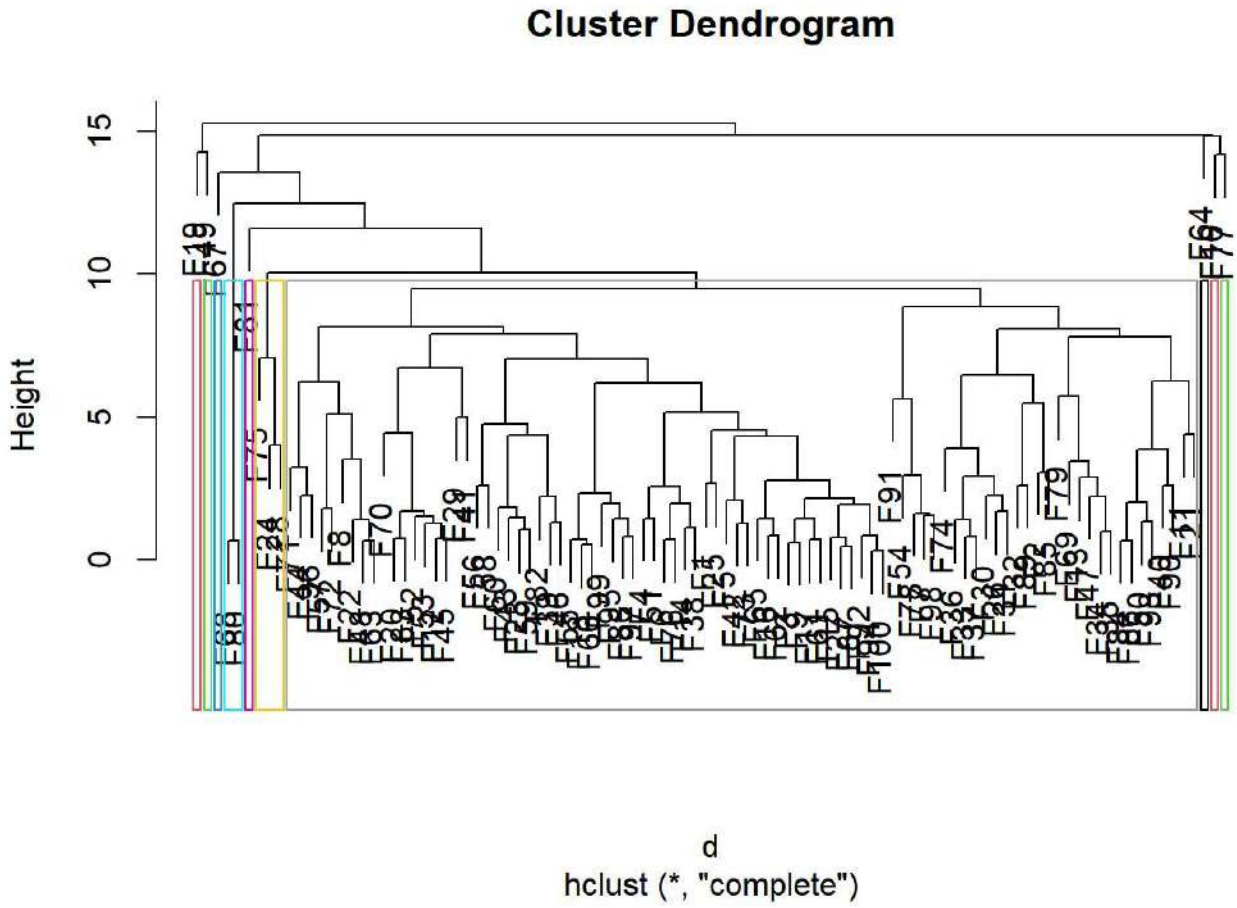
```
## [1] 0.2181529
```

Average Linkage apresenta índice silhouette de 0.22 e grupos também desproporcionais: 80, 1, 9, 1, 3, 1, 1, 1, 2 e 1.

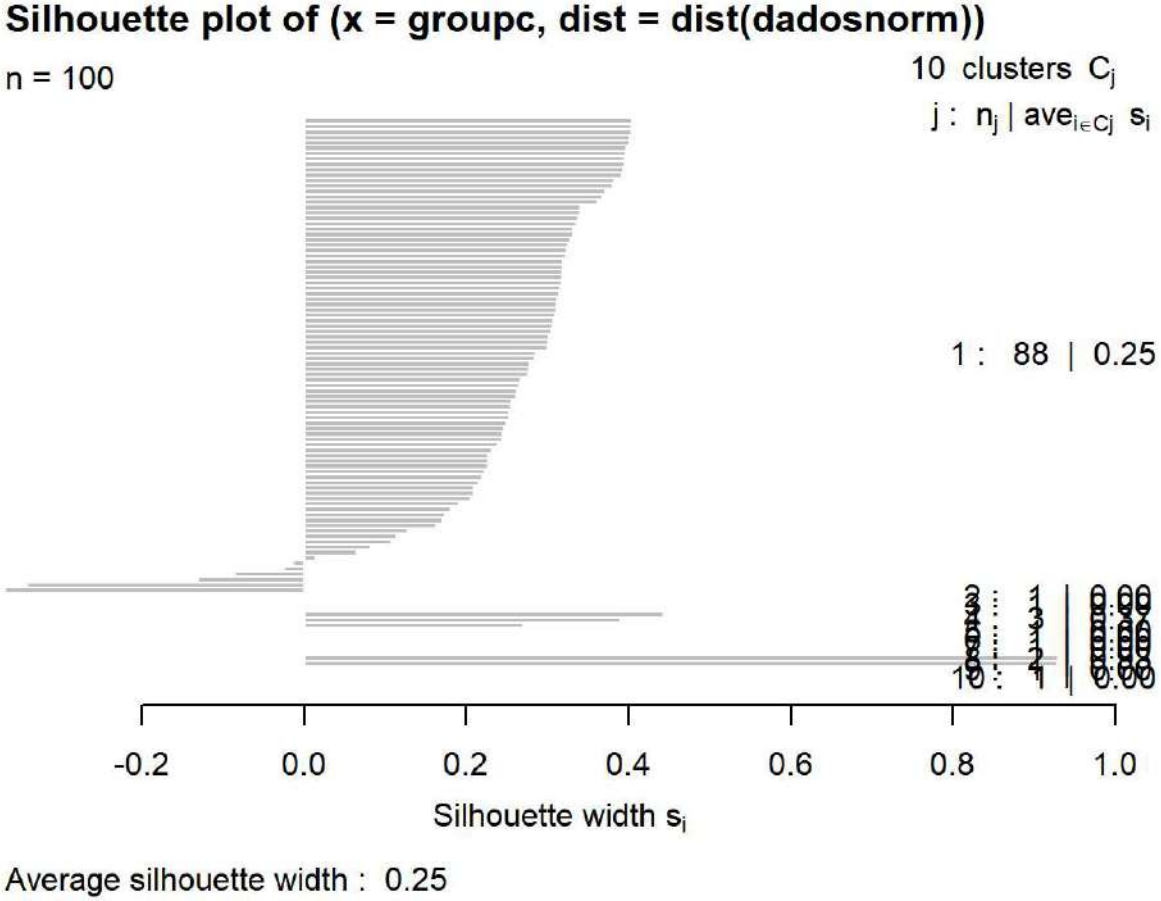
USANDO COMPLETE LINKAGE

```
d = dist(dadosnorm, method = "euclidean")
fitc = hclust(d, method = "complete")

par(mfrow=c(1,1))
plot(fitc)
groupc = cutree(fitc, k=10)
rect.hclust(fitc, k=10, border= 2:12)
```




```
ssco = silhouette(groupc, dist(dadosnorm))
plot(ssco)
```



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

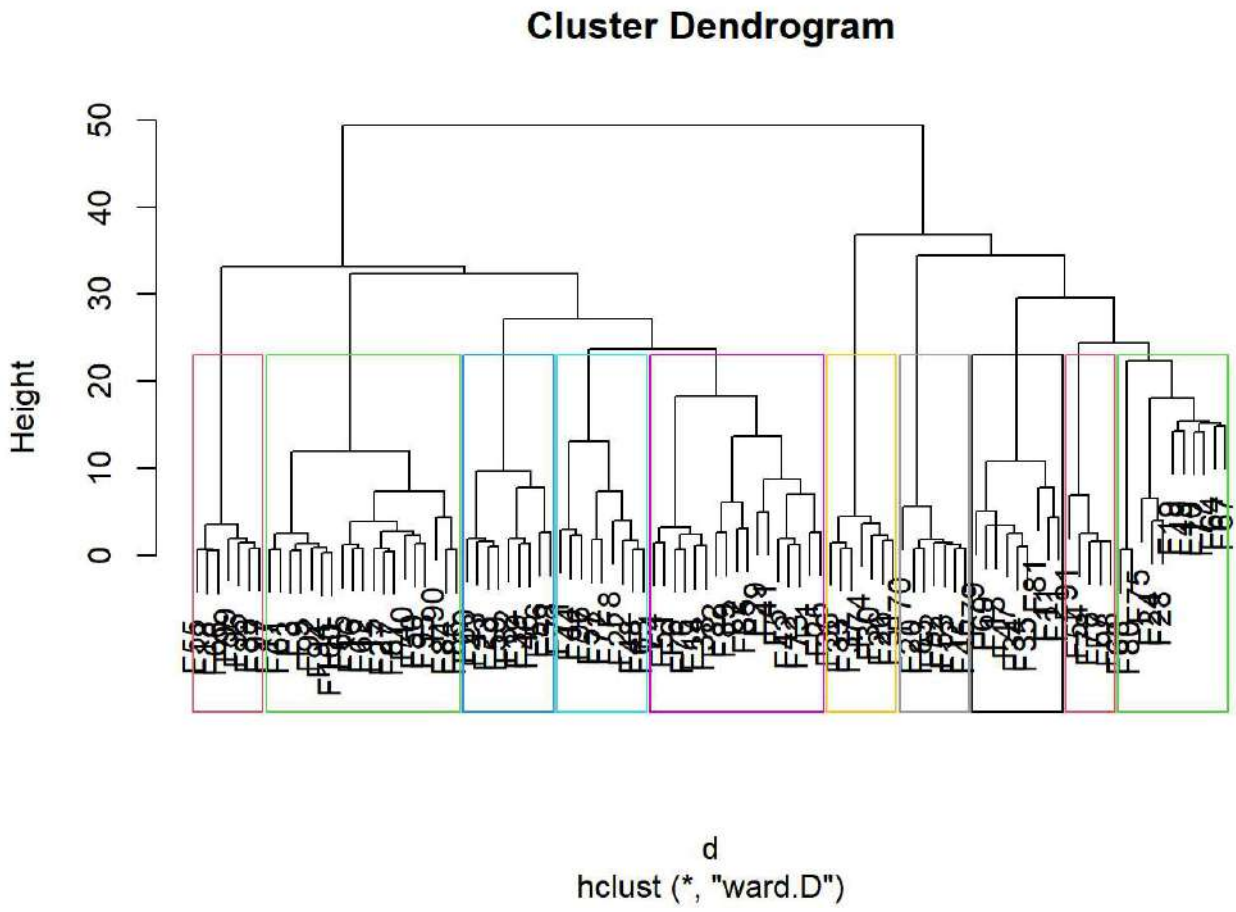
```
## [1] 0.251287
```

Complete Linkage apresenta índice silhouette de 0.25 e grupos também desproporcionais: 88, 1, 1, 3, 1, 1, 1, 2, 1 e 1.

USANDO WARD.D

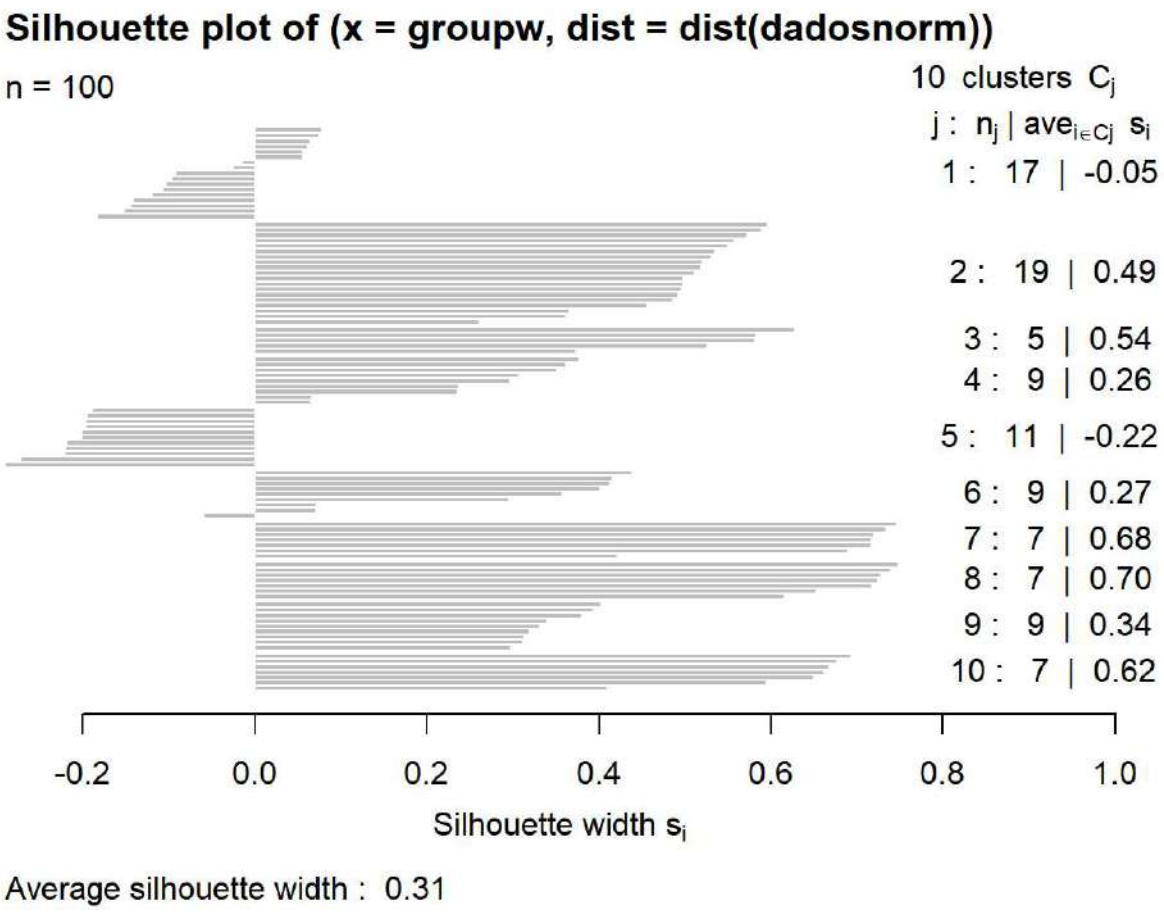
```
d = dist(dadosnorm, method = "euclidean")
fitw = hclust(d, method = "ward.D")

par(mfrow=c(1,1))
plot(fitw)
groupw = cutree(fitw, k=10)
rect.hclust(fitw, k=10, border= 2:12)
```



O dendrograma do ward.D mostra um certo equilíbrio na divisão dos 100 filmes entre os 10 grupos, o que não foi visto nos gráficos dos modelos anteriores.

```
sscw = silhouette(groupw, dist(dadosnorm))
plot(sscw)
```




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

```
## [1] 0.3072786
```

Ward.D apresenta índice silhouette de 0.31 e os grupos com distribuição mais proporcional dentre todos os métodos aplicados: 17, 19, 5, 9, 11, 9, 7, 7, 9 e 7.

PARTE 4: CONCLUSÃO

O melhor agrupamento de classes dos 100 filmes foi encontrado usando o Agrupamento Hierárquico (AH) com a medida **ward.D**, cujo índice Silhouette foi de **0.31**. O ranking final ficou assim:

- 1º) AH **Ward.D** = 0.31
- 2º) **K-Medóides** = 0.30
- 3º) AH **Complete Linkage** = 0.251
- 4º) **K-Médias** = 0.245
- 5º) AH **Average Linkage** = 0.22
- 6º) AH **Single Linkage** = 0.21

Combinando e salvando os resultados dos algoritmos com o arquivo original dos filmes da IMDb, para análise mais aprofundada dos agrupamentos e para descobrir padrões e insights que possam ser usados no processo de seleção e criação de conteúdo.

```
clusterizacao= cbind(novoimdb, Kmedias, kmedoides, groups, groupa, groupc, groupw)
```

```
write_delim(clusterizacao, "Agrupamentos dos 100 filmes pelos algoritmos de clusterização", d
elim = ";")
getwd()
```

```
## [1] "C:/Users/mmateus/Desktop/DATA-SCIENCE/Clustering-R"
```