

Análise e classificação de sementes de abóbora

Marcos Perazo Viana

28 de setembro de 2025

Ambiente do RStudio

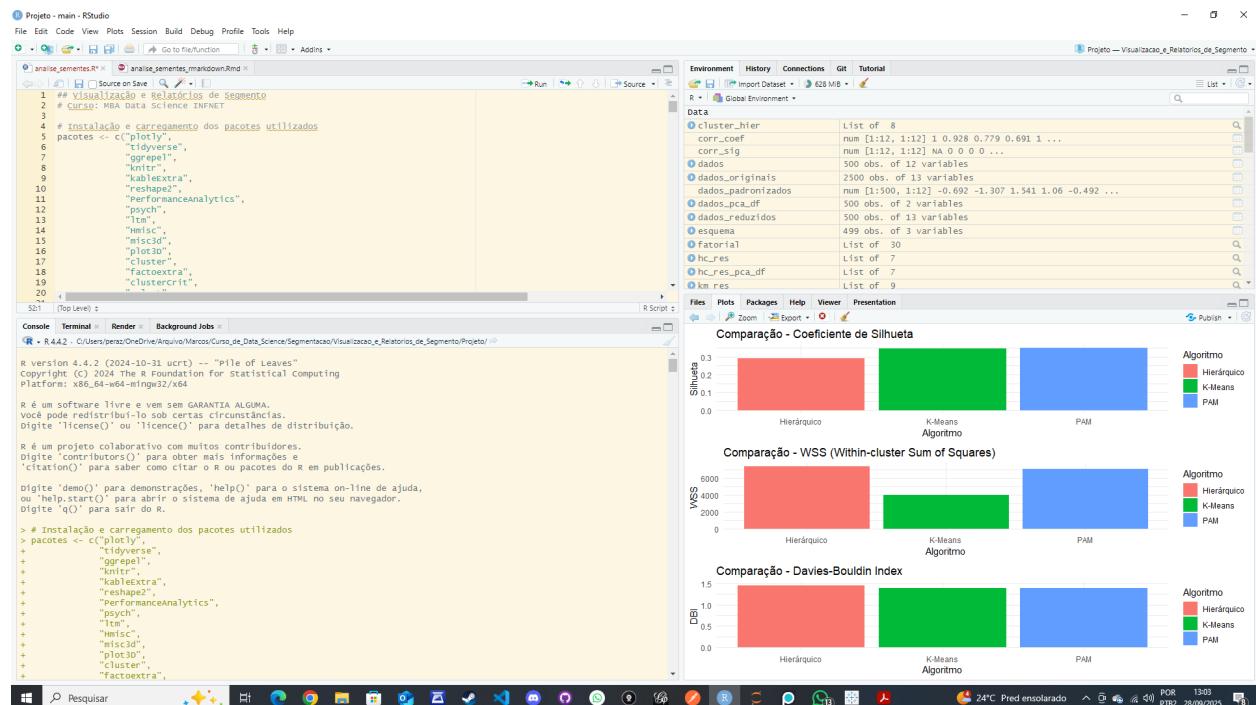


Figure 1: Ambiente RStudio

A base escolhida

A base de dados escolhida se encontra no endereço Pumpkin Seeds Dataset. Ela foi criada para classificação de sementes com base em características morfológicas extraídas de imagens. Isso tem aplicações diretas em: Agricultura de precisão, estudos botânicos e em processos industriais de seleção de sementes.

```
dados_originais <- read_excel("Pumpkin_Seeds_Dataset.xlsx")
```

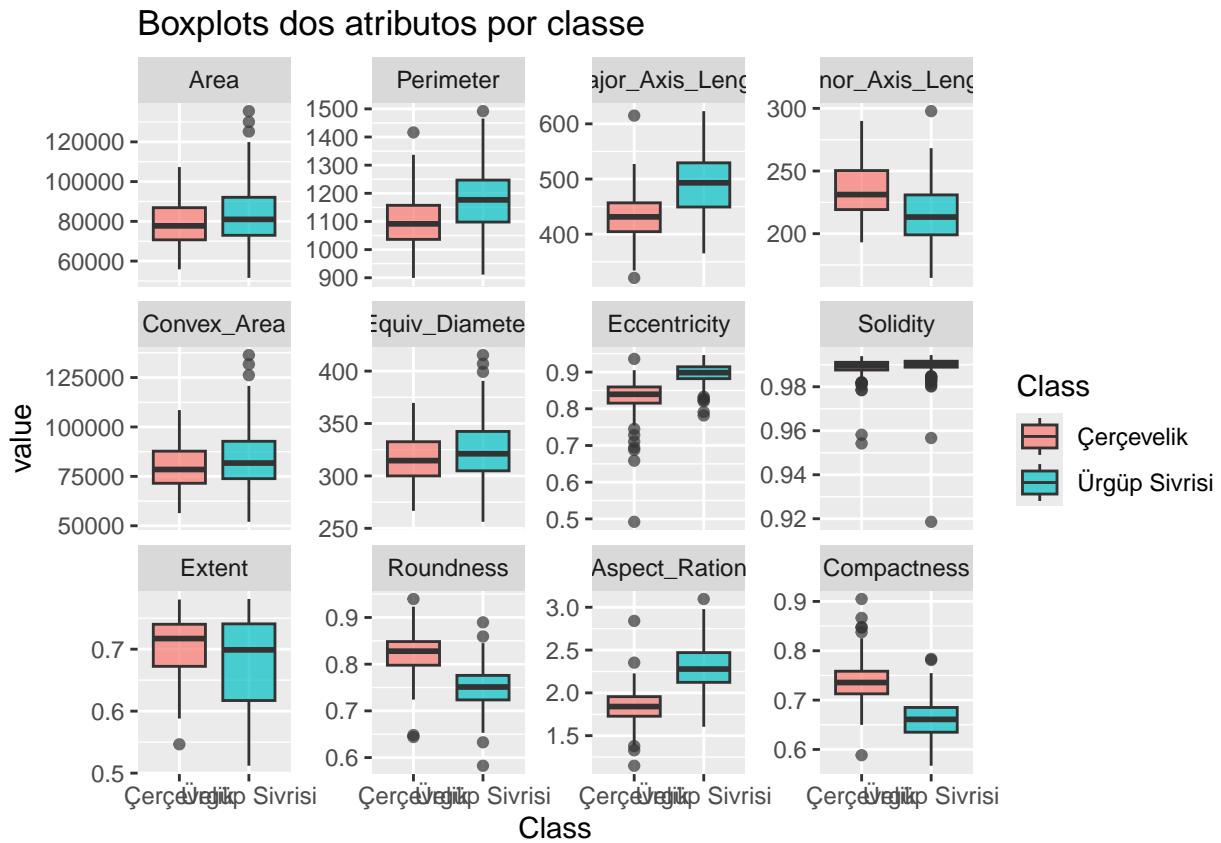
```
set.seed(223)
dados_reduzidos <- dados_originais %>% slice_sample(n = 500)
dados_reduzidos <- as.data.frame(dados_reduzidos)
```

```

dados_long <- melt(dados_reduzidos, id.vars = "Class")

ggplot(dados_long, aes(x=Class, y=value, fill=Class)) +
  geom_boxplot(alpha=0.7) +
  facet_wrap(~ variable, scales='free_y') +
  labs(title = "Boxplots dos atributos por classe")

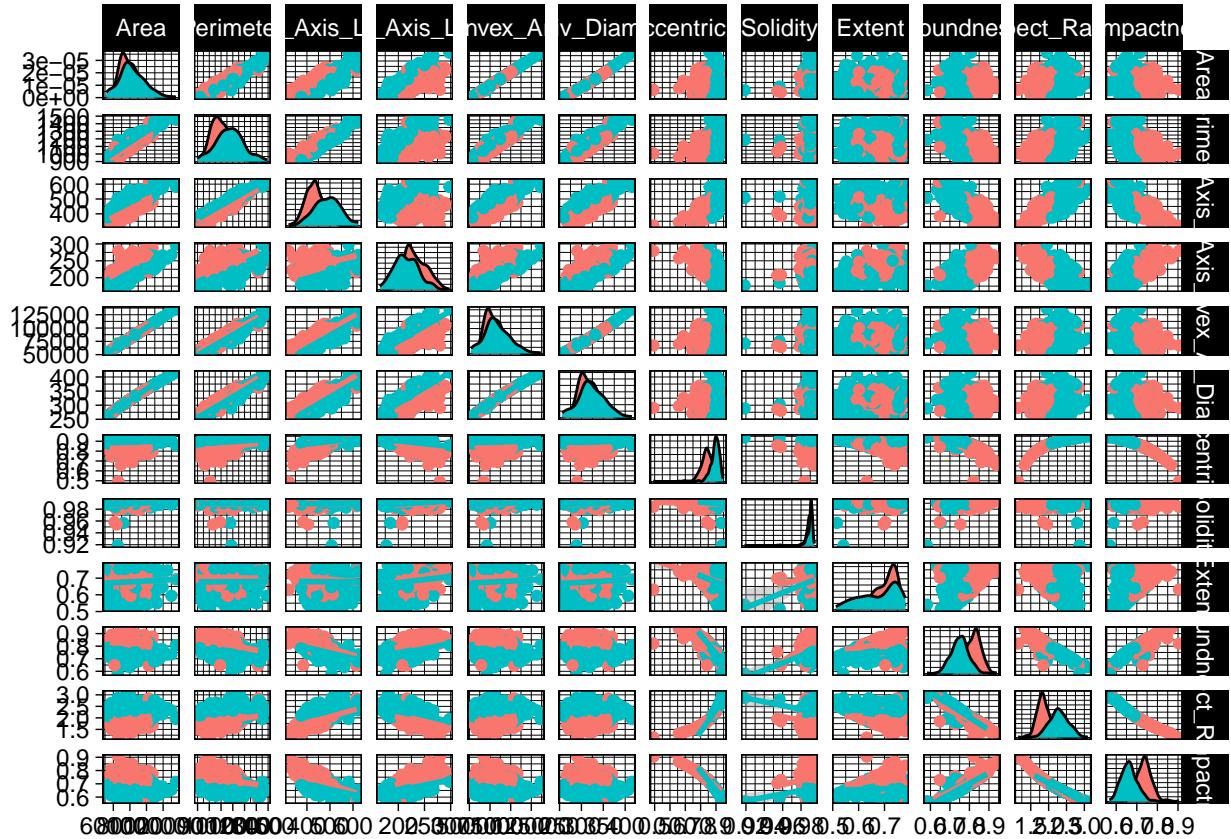
```



```

ggpairs(dados_reduzidos,
        columns = 1:12,      # Apenas variáveis numéricas
        aes(color = Class), # Cor por classe
        upper = list(continuous = "points"),
        lower = list(continuous = "smooth"),
        diag = list(continuous = "densityDiag")) +
  theme_linedraw()

```



```
dados <- dados_reduzidos %>% dplyr::select(-Class)
```

```
knitr::kable(head(dados), caption = "Primeiras linhas do conjunto de dados de sementes de abóbora")
```

Table 1: Primeiras linhas do conjunto de dados de sementes de abóbora

Area	Perimeter	Major_Axis_Length	Minor_Axis_Length	Convex_Area	Equiv_Diameter	Eccentricity	Solidity	Extent	Roundness	Aspect_Ration	Compactness
60925	957.318	375.9761	207.4241	61599	278.5177	0.8340	0.9891	0.7122	0.8354	1.8126	0.7408
83832	1135.230	448.0772	239.1491	84719	326.7082	0.8457	0.9895	0.7613	0.8174	1.8736	0.7291
77070	1152.830	486.4784	202.5228	77800	313.2548	0.9092	0.9906	0.5715	0.7287	2.4021	0.6439
71614	1036.482	405.5690	226.0209	72260	301.9632	0.8303	0.9911	0.6897	0.8377	1.7944	0.7445
66259	1013.800	410.5810	206.1564	66874	290.4541	0.8648	0.9908	0.7326	0.8101	1.9916	0.7074
65787	1021.125	409.1759	205.6027	66648	289.4177	0.8646	0.9871	0.6096	0.7929	1.9901	0.7073

```
str(dados) # Estrutura das variáveis
```

```
## 'data.frame': 500 obs. of 12 variables:
## $ Area : num 60925 83832 77070 71614 66259 ...
## $ Perimeter : num 957 1135 1153 1036 1014 ...
## $ Major_Axis_Length: num 376 448 486 406 411 ...
## $ Minor_Axis_Length: num 207 239 203 226 206 ...
## $ Convex_Area : num 61599 84719 77800 72260 66874 ...
## $ Equiv_Diameter : num 279 327 313 302 290 ...
## $ Eccentricity : num 0.834 0.846 0.909 0.83 0.865 ...
## $ Solidity : num 0.989 0.99 0.99 0.99 0.991 ...
## $ Extent : num 0.712 0.761 0.572 0.69 0.733 ...
## $ Roundness : num 0.835 0.817 0.729 0.838 0.81 ...
## $ Aspect_Ration : num 1.81 1.87 2.4 1.79 1.99 ...
## $ Compactness : num 0.741 0.729 0.644 0.745 0.707 ...
```

```
summary(dados) # Estatísticas básicas
```

```
##      Area      Perimeter   Major_Axis_Length   Minor_Axis_Length
## Min. : 51555  Min. : 899.5  Min. :320.8    Min. :164.7
## 1st Qu.: 71621  1st Qu.:1058.3  1st Qu.:419.3   1st Qu.:210.8
## Median : 79195  Median :1126.2  Median :449.3   Median :224.4
## Mean   : 81013  Mean   :1134.0  Mean   :458.3   Mean   :226.0
## 3rd Qu.: 89286  3rd Qu.:1202.9  3rd Qu.:492.6   3rd Qu.:241.2
## Max.  :135455  Max.  :1491.9  Max.  :623.0   Max.  :297.8
```

```

## Convex_Area      Equiv_Diameter    Eccentricity      Solidity
## Min.   : 52013   Min.   :256.2       Min.   :0.4921     Min.   :0.9186
## 1st Qu.: 72482   1st Qu.:302.0      1st Qu.:0.8338     1st Qu.:0.9880
## Median : 79997   Median :317.5      Median :0.8649     Median :0.9901
## Mean   : 81892   Mean   :320.1      Mean   :0.8617     Mean   :0.9891
## 3rd Qu.: 90187   3rd Qu.:337.2      3rd Qu.:0.8959     3rd Qu.:0.9914
## Max.  :136373   Max.  :415.3      Max.  :0.9464     Max.  :0.9944
## Extent   Roundness    Aspect_Ration    Compactness
## Min.   :0.5119   Min.   :0.5825     Min.   :1.149      Min.   :0.5670
## 1st Qu.:0.6526   1st Qu.:0.7528     1st Qu.:1.812      1st Qu.:0.6651
## Median :0.7119   Median :0.7931     Median :1.992      Median :0.7056
## Mean   :0.6920   Mean   :0.7899     Mean   :2.048      Mean   :0.7030
## 3rd Qu.:0.7403   3rd Qu.:0.8312     3rd Qu.:2.251      3rd Qu.:0.7414
## Max.  :0.7810   Max.  :0.9396     Max.  :3.097      Max.  :0.9049

```

O data frame dados possui 16 variáveis numéricas que descrevem características morfológicas das sementes de abóbora, como área, perímetro, redondeza e razão de aspecto. Essas variáveis serão utilizadas para classificar os tipos de sementes presentes no conjunto.

```
dados_padronizados <- scale(dados)
```

```

ground_truth <- dados_reduzidos$Class
n_rows <- nrow(dados_padronizados)
k_clusters <- 2 # numero de clusters

```

K-MEANS

```

set.seed(223)
kmeans_res <- kmeans(dados_padronizados, centers = k_clusters, nstart = 25)

```

PAM

```
pam_res <- pam(dados_padronizados, k = k_clusters)
```

HIERÁRQUICO

```

dist_matrix <- dist(dados_padronizados, method = "euclidean")
hc_res <- hclust(dist_matrix, method = "ward.D2") # linkage Ward
hc_clusters <- cutree(hc_res, k = k_clusters)

```

MÉTRICAS INTERNAS

Silhueta

```

sil_kmeans <- silhouette(kmeans_res$cluster, dist(dados_padronizados))
sil_pam <- silhouette(pam_res$clustering, dist(dados_padronizados))
sil_hc <- silhouette(hc_clusters, dist(dados_padronizados))

```

WSS (Within-cluster Sum of Squares)

```
wss_kmeans <- kmeans_res$tot.withinss
wss_pam <- sum(sapply(1:k_clusters, function(k) sum((dados_padronizados$pam_res$clustering == k, ] - pam_res$tot.withinss)))
wss_hc <- sum(sapply(1:k_clusters, function(k) {
  cluster_points <- dados_padronizados[hc_clusters == k, , drop = FALSE]
  center <- colMeans(cluster_points)
  sum(rowSums((cluster_points - center)^2))
}))
```

Davies-Bouldin

```
db_kmeans <- index.DB(dados_padronizados, kmeans_res$cluster)$DB
db_pam <- index.DB(dados_padronizados, pam_res$clustering)$DB
db_hc <- index.DB(dados_padronizados, hc_clusters)$DB
```

RESULTADOS

```
cat("Resultados (Métricas Internas):\n")
```

```
## Resultados (Métricas Internas):
```

```
cat("K-Means - Silhueta:", mean(sil_kmeans[, 3]),
  " WSS:", wss_kmeans,
  " DBI:", db_kmeans, "\n")
```

```
## K-Means - Silhueta: 0.305765 WSS: 4119.997 DBI: 1.474202
```

```
cat("PAM      - Silhueta:", mean(sil_pam[, 3]),
  " WSS:", wss_pam,
  " DBI:", db_pam, "\n")
```

```
## PAM      - Silhueta: 0.3124865 WSS: 7407.68 DBI: 1.456355
```

```
cat("Hierárq - Silhueta:", mean(sil_hc[, 3]),
  " WSS:", wss_hc,
  " DBI:", db_hc, "\n")
```

```
## Hierárq - Silhueta: 0.2857827 WSS: 7343.582 DBI: 1.557811
```

```
resultados <- data.frame(
  Algoritmo = c("K-Means", "PAM", "Hierárquico"),
  Silhueta = c(mean(sil_kmeans[, 3]), mean(sil_pam[, 3]), mean(sil_hc[, 3])),
  WSS = c(wss_kmeans, wss_pam, wss_hc),
  DBI = c(db_kmeans, db_pam, db_hc)
)

print(resultados)
```

```

##      Algoritmo   Silhueta      WSS       DBI
## 1      K-Means 0.3057650 4119.997 1.474202
## 2          PAM 0.3124865 7407.680 1.456355
## 3 Hierárquico 0.2857827 7343.582 1.557811

```

Criação do Gráfico com os resultados

```

p1 <- ggplot(resultados, aes(x = Algoritmo, y = Silhueta, fill = Algoritmo)) +
  geom_bar(stat = "identity") +
  ggtitle("Comparação – Coeficiente de Silhueta") +
  theme_minimal()

```

```

p2 <- ggplot(resultados, aes(x = Algoritmo, y = WSS, fill = Algoritmo)) +
  geom_bar(stat = "identity") +
  ggtitle("Comparação – WSS (Within-cluster Sum of Squares)") +
  theme_minimal()

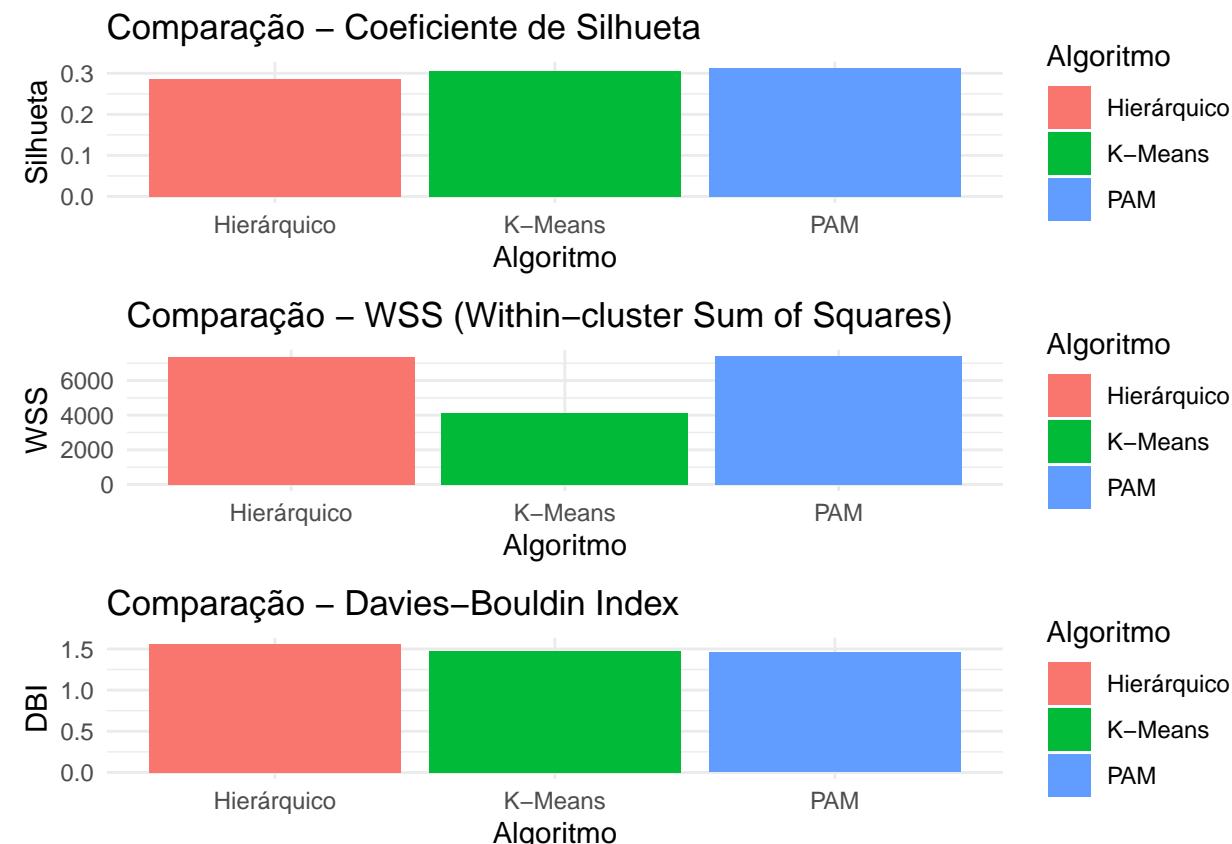
```

```

p3 <- ggplot(resultados, aes(x = Algoritmo, y = DBI, fill = Algoritmo)) +
  geom_bar(stat = "identity") +
  ggtitle("Comparação – Davies–Bouldin Index") +
  theme_minimal()

```

```
grid.arrange(p1, p2, p3, nrow = 3)
```



Nesse cenário, o algoritmo PAM apresentou o melhor desempenho geral, com o maior índice de silhueta e o menor DBI, indicando boa separação entre os grupos e baixa sobreposição. Apesar do WSS elevado, o modelo mostrou-se mais coeso e interpretável.

Aplicação do algoritmo PCA nos dados normalizados

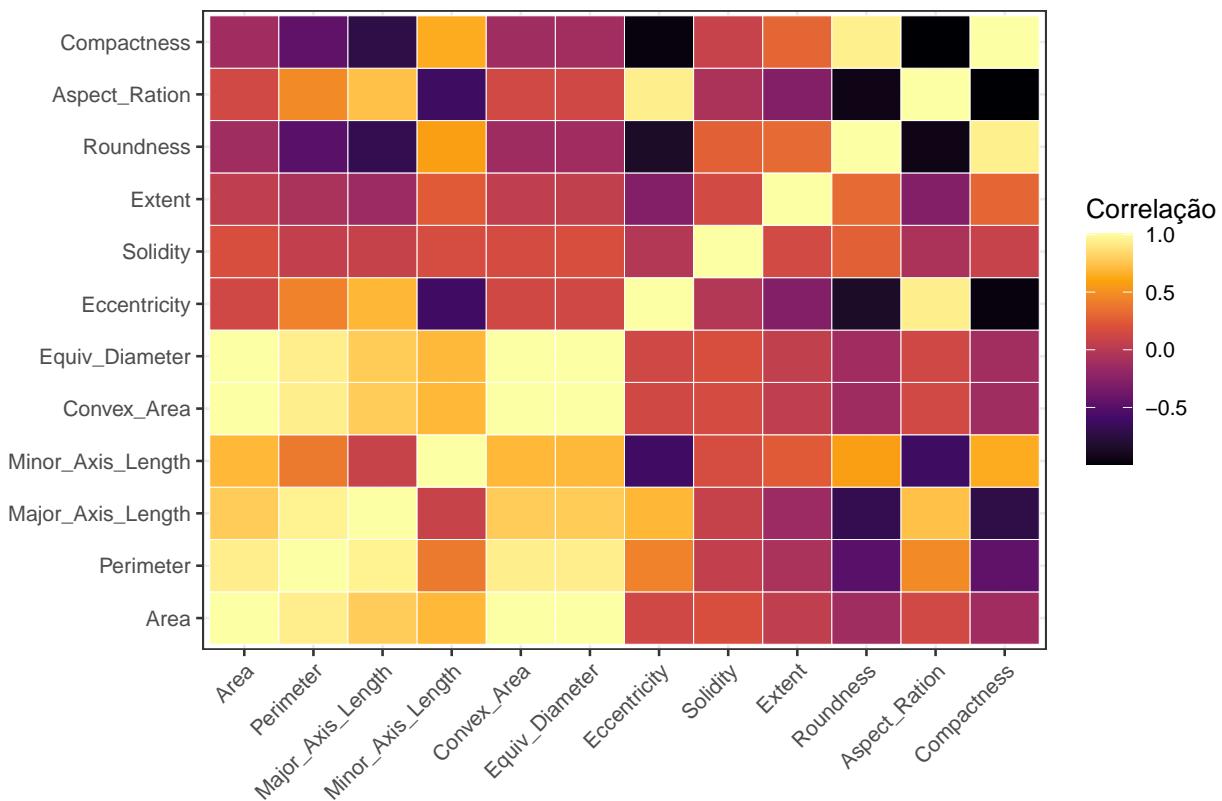
```
rho <- rcorr(as.matrix(dados_padronizados), type="pearson")
corr_coef <- rho$r           # Matriz de correlações
corr_sig <- round(rho$p, 5)  # Matriz com p-valor dos coeficientes

# Calcula correlação e transforma em formato longo
correlacoes <- cor(dados_padronizados)
cor_melt <- melt(correlacoes)

# Renomeia coluna de correlação
names(cor_melt) <- c("Var1", "Var2", "Correlacao")

# Mapa de calor com ggplot2
ggplot(cor_melt, aes(x = Var1, y = Var2, fill = Correlacao)) +
  geom_tile(color = "white") +
  scale_fill_viridis_c(option = "B", direction = 1) +
  labs(title = "Mapa de calor das correlações de Pearson",
       x = NULL, y = NULL, fill = "Correlação") +
  theme_bw(base_size = 10) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Mapa de calor das correlações de Pearson



```
cortest.bartlett(dados_padronizados)
```

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

## $chisq
## [1] 24698.48
##
## $p.value
## [1] 0
##
## $df
## [1] 66
```

Encontramos p.value é igual a zero

```
fatorial <- principal(dados_padronizados,
                      nfactors = ncol(dados_padronizados),
                      rotate = "none",
                      scores = TRUE)
cum_var <- fatorial$Vaccounted["Cumulative Var", ]
```

```
##      PC1       PC2       PC3       PC4       PC5       PC6       PC7       PC8
## 0.4782643 0.8291104 0.9158666 0.9856008 0.9955125 0.9991007 0.9995176 0.9998248
##      PC9       PC10      PC11      PC12
## 0.9999793 0.9999943 0.9999996 1.0000000
```

```

num_comp <- which(cum_var >= 0.7) [1]
num_comp

## PC2
##    2

variancia_compartilhada <- as.data.frame(fatorial$Vaccounted) %>%
  slice(1:3)

rownames(variancia_compartilhada) <- c("Autovalores",
                                         "Prop. da Variância",
                                         "Prop. da Variância Acumulada")

round(variancia_compartilhada, 3) %>%
  kable() %>%
  kable_styling(bootstrap_options = "striped",
                full_width = FALSE,
                font_size = 20)

```

	PC1	PC2	PC3	PC4	PC5	
Autovalores	5.739	4.210	1.041	0.837	0.119	0
Prop. da Variância	0.478	0.351	0.087	0.070	0.010	0
Prop. da Variância Acumulada	0.478	0.829	0.916	0.986	0.996	0

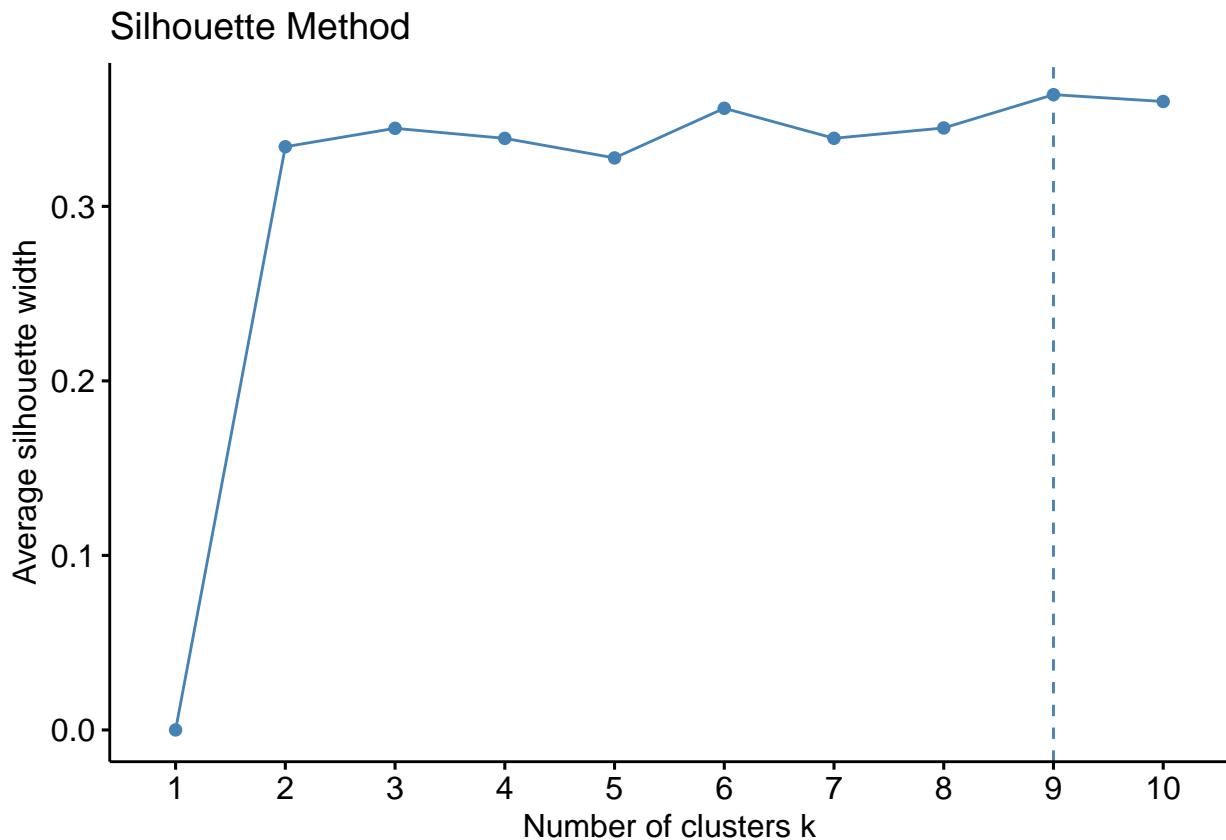
```

pca_df <- data.frame(
  fatorial$scores[, 1:2],
  classe_da_semente = dados_reduzidos$Class
)

pca_df_dados <- pca_df %>% dplyr::select(-classe_da_semente)

fviz_nbclust(pca_df_dados, kmeans, method = "silhouette") +
  ggplot2::labs(title = "Silhouette Method")

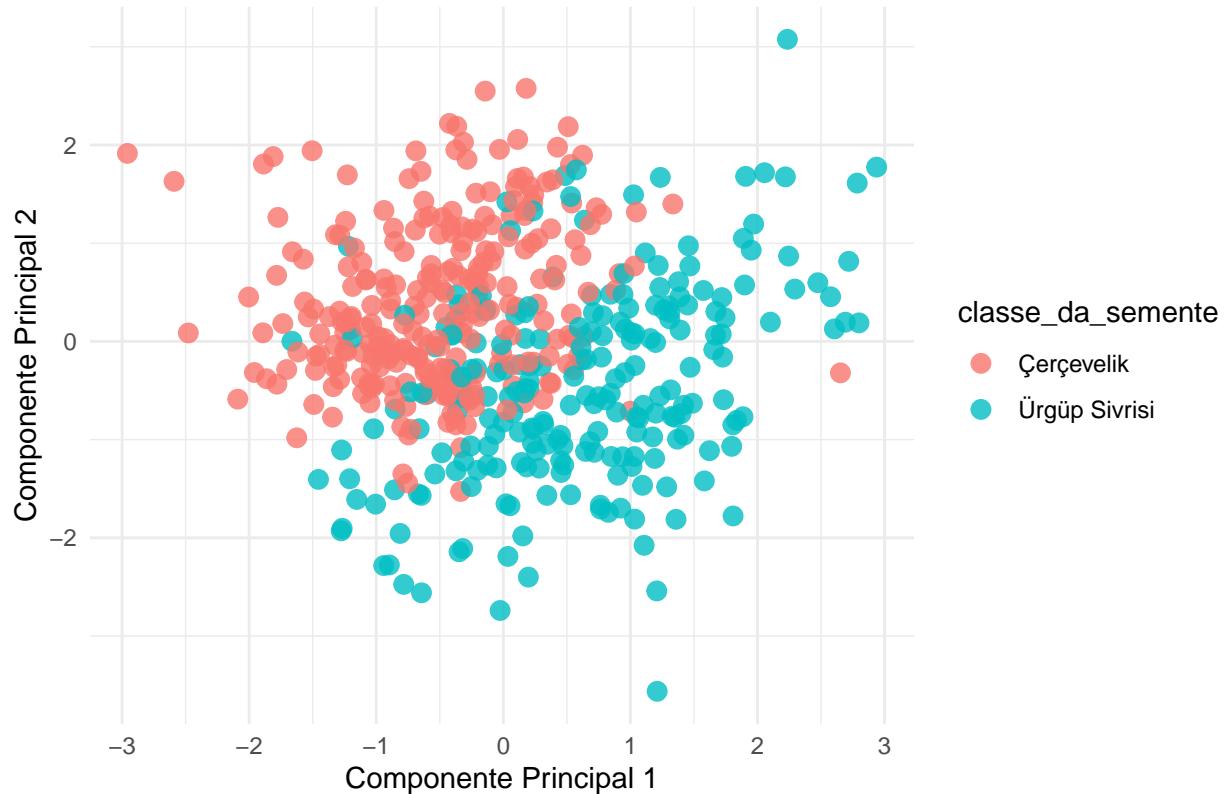
```



O número de centróides pode ser obtido utilizando o índice de Silhueta. No gráfico acima, podemos verificar que são determinados nove centróides. Porém esse valor não condiz com os dados reais, que são apenas dois tipos de sementes diferentes.

```
ggplot(pca_df, aes(x = PC1, y = PC2, color = classe_da_semente)) +
  geom_point(size = 3, alpha = 0.8) +
  labs(title = "PCA - Classe da Semente",
       x = "Componente Principal 1",
       y = "Componente Principal 2") +
  theme_minimal()
```

PCA – Classe da Semente

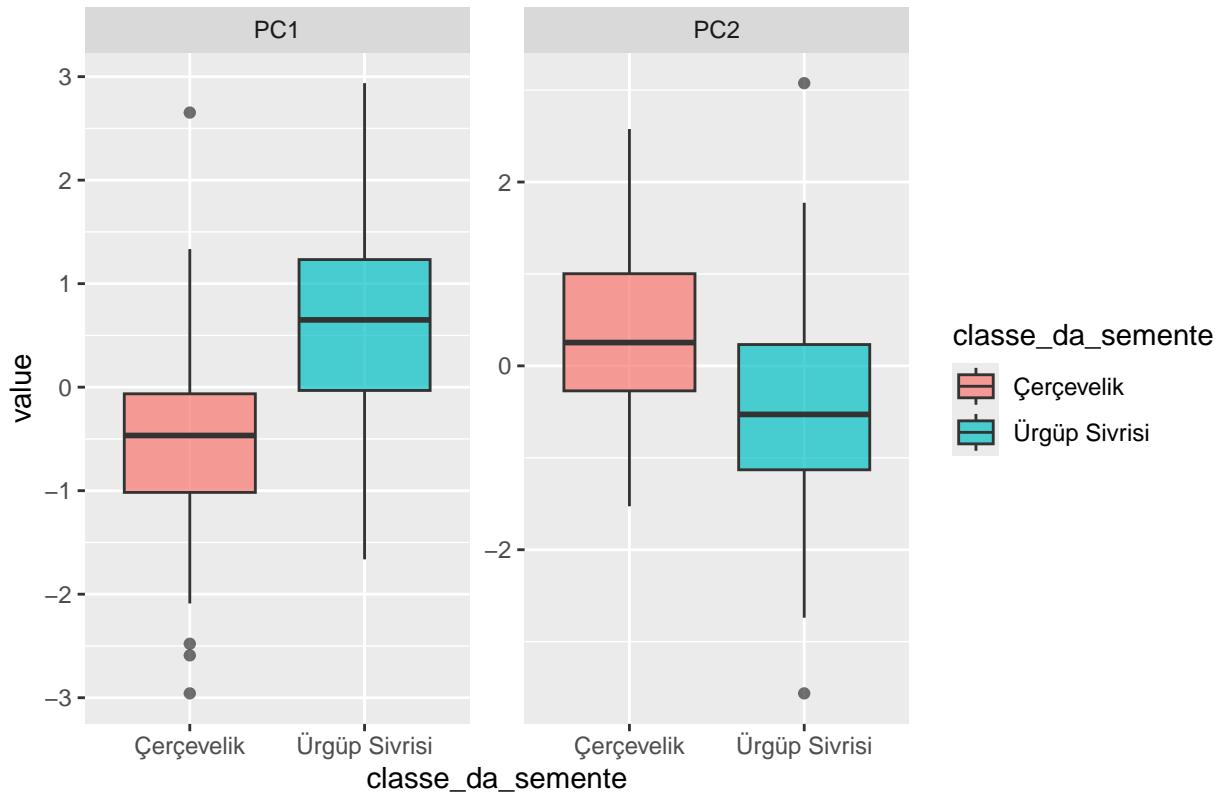


Análise dos dados após PCA

```
pca_df_long <- melt(pca_df, id.vars = "classe_da_semente")

ggplot(pca_df_long, aes(x=classe_da_semente, y=value, fill=classe_da_semente)) +
  geom_boxplot(alpha=0.7) +
  facet_wrap(~ variable, scales='free_y') +
  labs(title = "Boxplots dos componentes gerados no PCA")
```

Boxplots dos componentes gerados no PCA



```
dados_pca_df <- pca_df %>% dplyr::select(-classe_da_semente)
```

```
knitr::kable(head(pca_df), caption = "Primeiras linhas do conjunto de dados de sementes de abóbora após o PCA")
```

Table 3: Primeiras linhas do conjunto de dados de sementes de abóbora após o PCA

	PC1	PC2	classe_da_semente
-1.4913049	-0.6421392	Çerçevelek	
-0.2311063	0.6059733	Çerçevelek	
0.6469567	-1.1201075	Ürgüp Sivrisi	
-0.9808628	0.0466117	Çerçevelek	
-0.8507799	-0.6896804	Ürgüp Sivrisi	
-0.7463888	-0.9529656	Çerçevelek	

```
str(dados_pca_df) # Estrutura das variáveis
```

```
## 'data.frame': 500 obs. of 2 variables:
## $ PC1: num -1.491 -0.231 0.647 -0.981 -0.851 ...
## $ PC2: num -0.6421 0.606 -1.1201 0.0466 -0.6897 ...
```

```
summary(dados_pca_df)      # Estatísticas básicas
```

```
##          PC1             PC2
##  Min.   :-2.9583   Min.   :-3.56295
##  1st Qu.:-0.6791   1st Qu.:-0.60628
##  Median :-0.1248   Median :-0.01365
##  Mean    : 0.0000   Mean   : 0.00000
##  3rd Qu.: 0.6484   3rd Qu.: 0.62758
##  Max.    : 2.9372   Max.   : 3.07587
```

```
ground_truth_pca_df <- pca_df$classe_da_semente
n_rows_pca_df <- nrow(dados_pca_df)
k_clusters_pca_df <- 2 # numero de clusters
```

K-MEANS

```
set.seed(223)
kmeans_res_pca_df <- kmeans(dados_pca_df, centers = k_clusters, nstart = 25)
```

PAM

```
pam_res_pca_df <- pam(dados_pca_df, k = k_clusters)
```

HIERÁRQUICO

```
dist_matrix_pca_df <- dist(dados_pca_df, method = "euclidean")
hc_res_pca_df <- hclust(dist_matrix_pca_df, method = "ward.D2") # linkage Ward
hc_clusters_pca_df <- cutree(hc_res_pca_df, k = k_clusters)
```

MÉTRICAS INTERNAS

Silhueta

```
sil_kmeans_pca_df <- silhouette(kmeans_res_pca_df$cluster, dist(dados_pca_df))
sil_pam_pca_df <- silhouette(pam_res_pca_df$clustering, dist(dados_pca_df))
sil_hc_pca_df <- silhouette(hc_clusters_pca_df, dist(dados_pca_df))
```

WSS (Within-cluster Sum of Squares)

```
wss_kmeans_pca_df <- kmeans_res_pca_df$tot.withinss
wss_pam_pca_df <- sum(sapply(1:k_clusters_pca_df, function(k) sum((dados_pca_df[pam_res_pca_df$clusterid == k, ] - center[k, ])^2)))
wss_hc_pca_df <- sum(sapply(1:k_clusters_pca_df, function(k) {
  cluster_points <- dados_pca_df[hc_clusters == k, , drop = FALSE]
  center <- colMeans(cluster_points)
  sum(rowSums((cluster_points - center)^2))
}))
```

Davies-Bouldin

```
db_kmeans_pca_df <- index.DB(dados_pca_df, kmeans_res_pca_df$cluster)$DB
db_pam_pca_df <- index.DB(dados_pca_df, pam_res_pca_df$clustering)$DB
db_hc_pca_df <- index.DB(dados_pca_df, hc_clusters_pca_df)$DB
```

RESULTADOS

```
cat("Resultados (Métricas Internas):\n")
```

```
## Resultados (Métricas Internas):
```

```
cat("K-Means - Silhueta:", mean(sil_kmeans_pca_df[, 3]),
  " WSS:", wss_kmeans_pca_df,
  " DBI:", db_kmeans_pca_df, "\n")
```

```
## K-Means - Silhueta: 0.3289184 WSS: 662.7697 DBI: 1.405934
```

```
cat("PAM      - Silhueta:", mean(sil_pam_pca_df[, 3]),
  " WSS:", wss_pam_pca_df,
  " DBI:", db_pam_pca_df, "\n")
```

```
## PAM      - Silhueta: 0.320953 WSS: 801.265 DBI: 1.429305
```

```
cat("Hierárq - Silhueta:", mean(sil_hc_pca_df[, 3]),
  " WSS:", wss_hc_pca_df,
  " DBI:", db_hc_pca_df, "\n")
```

```
## Hierárq - Silhueta: 0.3116495 WSS: 1212.051 DBI: 1.420029
```

```
resultados_pca_df <- data.frame(
  Algoritmo = c("K-Means", "PAM", "Hierárquico"),
  Silhueta = c(mean(sil_kmeans_pca_df[, 3]), mean(sil_pam_pca_df[, 3]), mean(sil_hc_pca_df[, 3])),
  WSS = c(wss_kmeans_pca_df, wss_pam_pca_df, wss_hc_pca_df),
  DBI = c(db_kmeans_pca_df, db_pam, db_hc_pca_df)
)

print(resultados_pca_df)
```

```

##      Algoritmo Silhueta      WSS      DBI
## 1      K-Means 0.3289184 662.7697 1.405934
## 2          PAM 0.3209530 801.2650 1.456355
## 3 Hierárquico 0.3116495 1212.0511 1.420029

```

Criação do Gráfico com os resultados após o PCA

```

p4 <- ggplot(resultados_pca_df, aes(x = Algoritmo, y = Silhueta, fill = Algoritmo)) +
  geom_bar(stat = "identity") +
  ggtitle("Comparação - Coeficiente de Silhueta") +
  theme_minimal()

```

```

p5 <- ggplot(resultados, aes(x = Algoritmo, y = WSS, fill = Algoritmo)) +
  geom_bar(stat = "identity") +
  ggtitle("Comparação - WSS (Within-cluster Sum of Squares)") +
  theme_minimal()

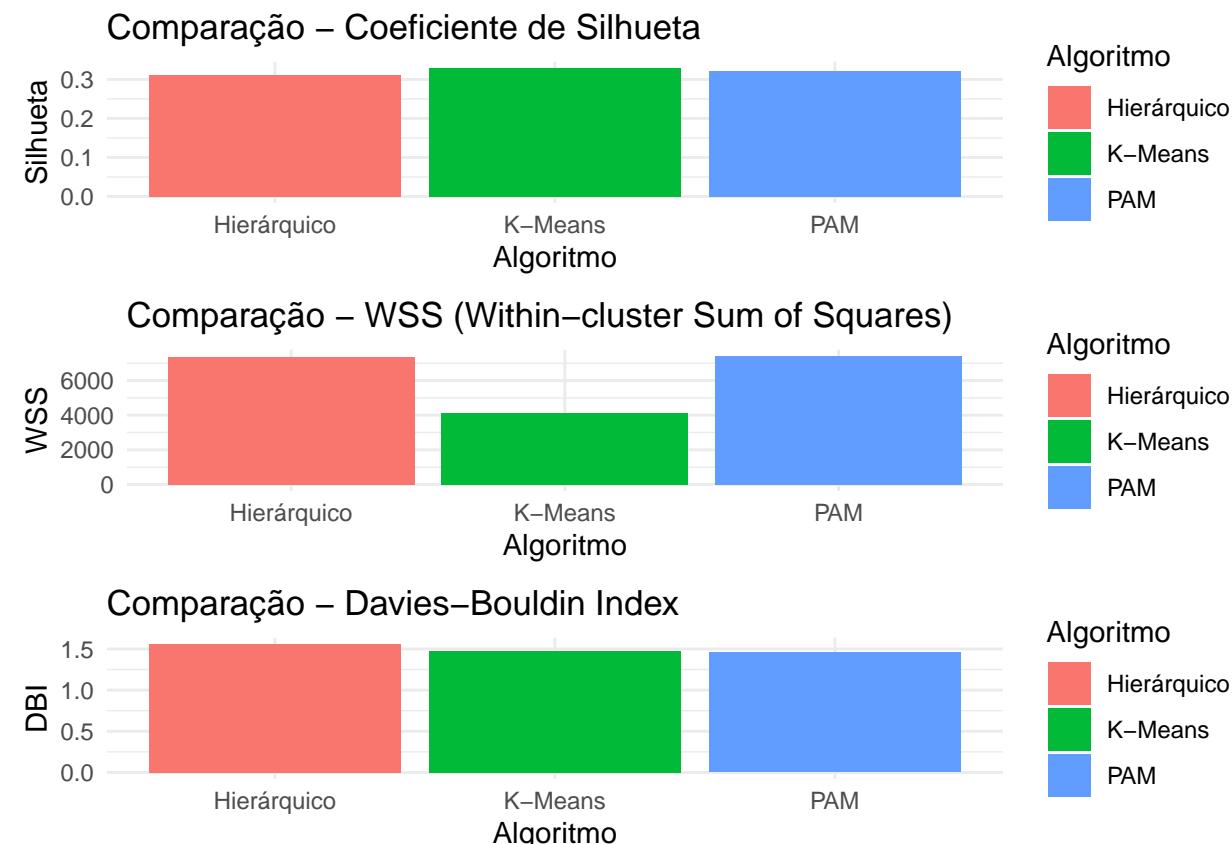
```

```

p6 <- ggplot(resultados, aes(x = Algoritmo, y = DBI, fill = Algoritmo)) +
  geom_bar(stat = "identity") +
  ggtitle("Comparação - Davies-Bouldin Index") +
  theme_minimal()

```

```
grid.arrange(p4, p5, p6, nrow = 3)
```



Após a aplicação do PCA, que reduziu a dimensionalidade dos dados mantendo a maior parte da variância explicada, observou-se uma melhora significativa nos resultados dos modelos. Com a nova representação dos dados, o algoritmo K-Means passou a liderar em todas as métricas, apresentando maior coesão interna (menor WSS), melhor separação entre os clusters (maior silhueta) e menor sobreposição (menor DBI). Isso evidencia o impacto positivo da redução de dimensionalidade na performance dos modelos de agrupamento.

Tableau

Segue o link do gráfico solicitado na sexta questão: Gráfico salvo no Tableau Server e o print do ambiente

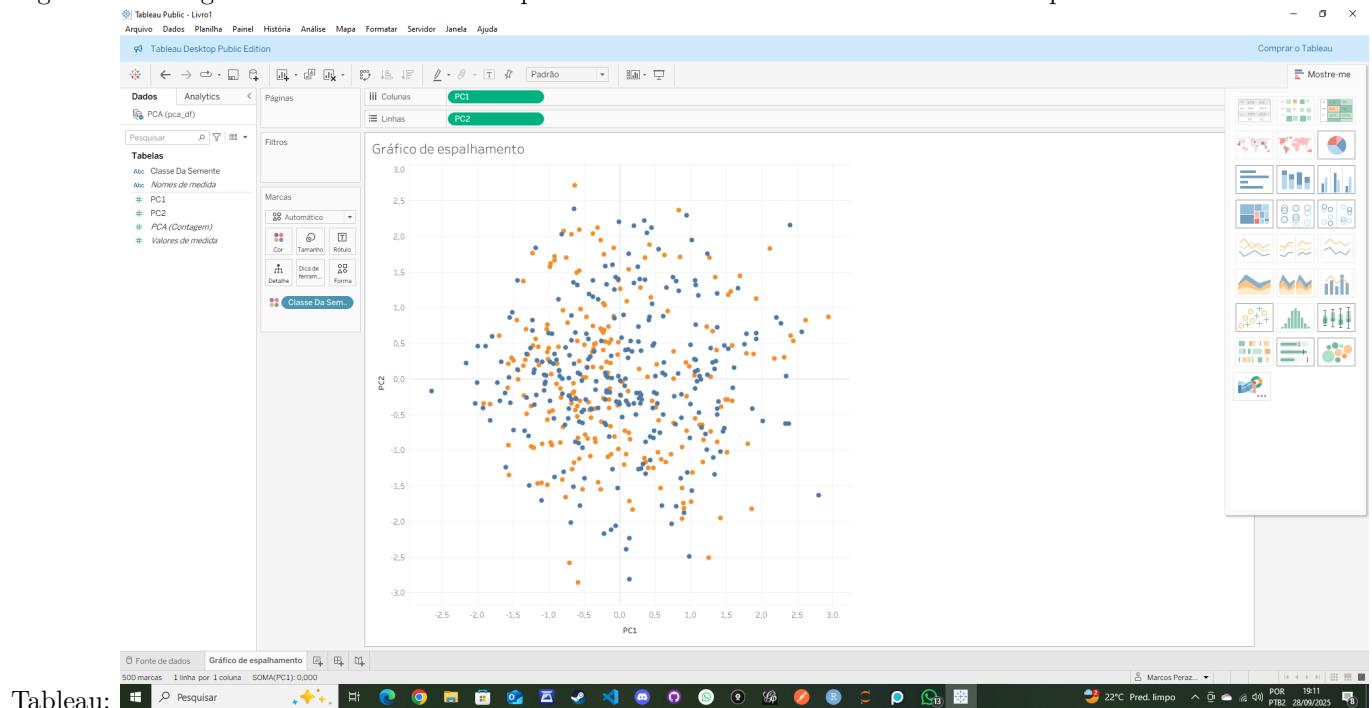


Tableau: