Analisi del Dataset Heart of catdata

Roberta Fusco & Luca Ruocco

2025-03-04

Il dataset heart disponibile nel pacchetto catdata in R è un insieme di dati utilizzato per studiare i fattori di rischio della malattia coronarica.

Questo dataset è stato raccolto da uno studio epidemiologico condotto nel Western Cape, Sud Africa, ed è ampiamente usato per l'analisi statistica e l'apprendimento automatico nel contesto medico.

Numero di osservazioni: 462 pazienti Numero di variabili: 10

Il dataset contiene una variabile binaria (target) e diverse variabili continue e categoriali.

y –> Binaria (0/1): Presenza di malattia coronarica (1 = sì, 0 = no)

sbp –> Numerica: Pressione arteriosa sistolica (mmHg)

tobacco –> Numerica: Consumo cumulativo di tabacco (kg)

ldl –> Numerica: Colesterolo lipoproteico a bassa densità (mg/dL)

adiposity –> Numerica: Adiposità corporea (percentuale)

famhist –> Categorica (0/1): Storia familiare di malattie cardiache (1 = presente, 0 = assente) typea –> Numerica: Indice di comportamento di personalità di tipo A (scala psicologica)

obesity –> Numerica: Indice di obesità (derivato da peso e altezza)

alcohol –> Numerica: Consumo di alcol (quantità media assunta)

age –> Numerica: Età del paziente (anni)

Obiettivo: Identificare i fattori di rischio associati alla malattia coronarica (CHD - Coronary Heart Disease).

```r
# Analisi Esplorativa
library(GGally)
library(ggplot2)

# Selezionare le variabili numeriche
numeric_vars <- c("sbp", "tobacco", "ldl", "adiposity", "typea",
                  "obesity", "alcohol", "age")

# Creare il dataframe con solo le variabili numeriche e la variabile target
heart_numeric <- heart[, numeric_vars]
heart_numeric$y <- as.factor(heart$y)

# Creazione della tabella di frequenza
counts <- table(heart$y)

# Calcolo delle percentuali
percentages <- round(100 * counts / sum(counts), 1)
```

```r
# Creazione del barplot con le percentuali
barplot(counts,
        main = "Distribuzione della Malattia Cardiaca",
        col = c("blue", "red"),
        ylim = c(0, max(counts) * 1.2),
        names.arg = c("No Malattia", "Malattia"))

# Aggiunta delle etichette di percentuale sopra le barre
text(x = seq_along(counts),
     y = counts,
     label = paste0(percentages, "%"),
     pos = 3,
     col = "black",
     cex = 1.2)
```

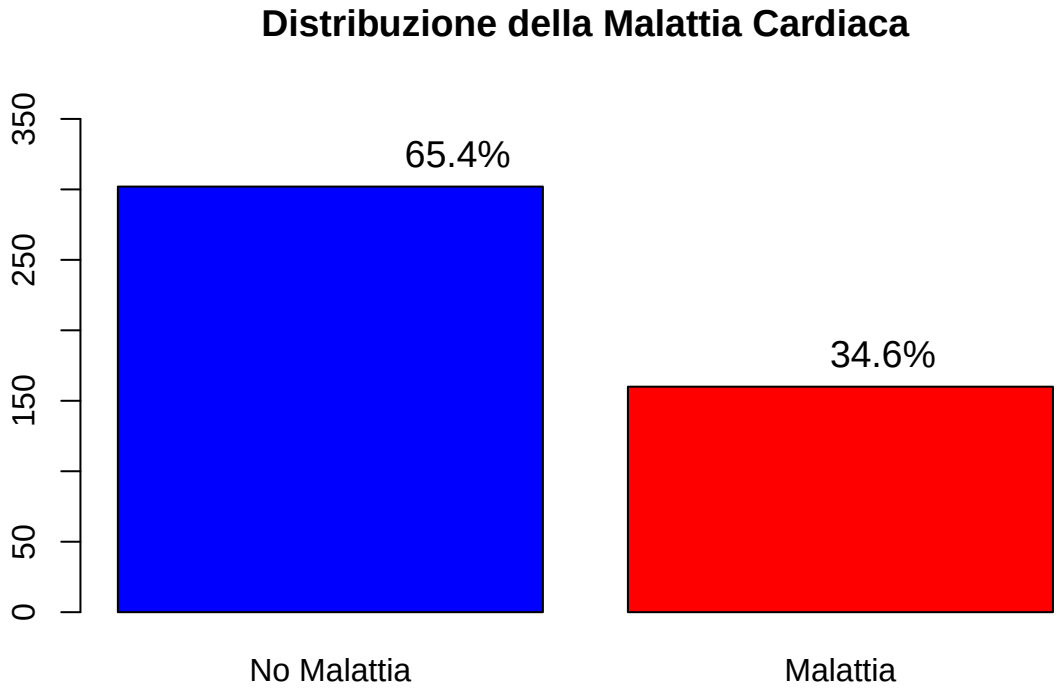
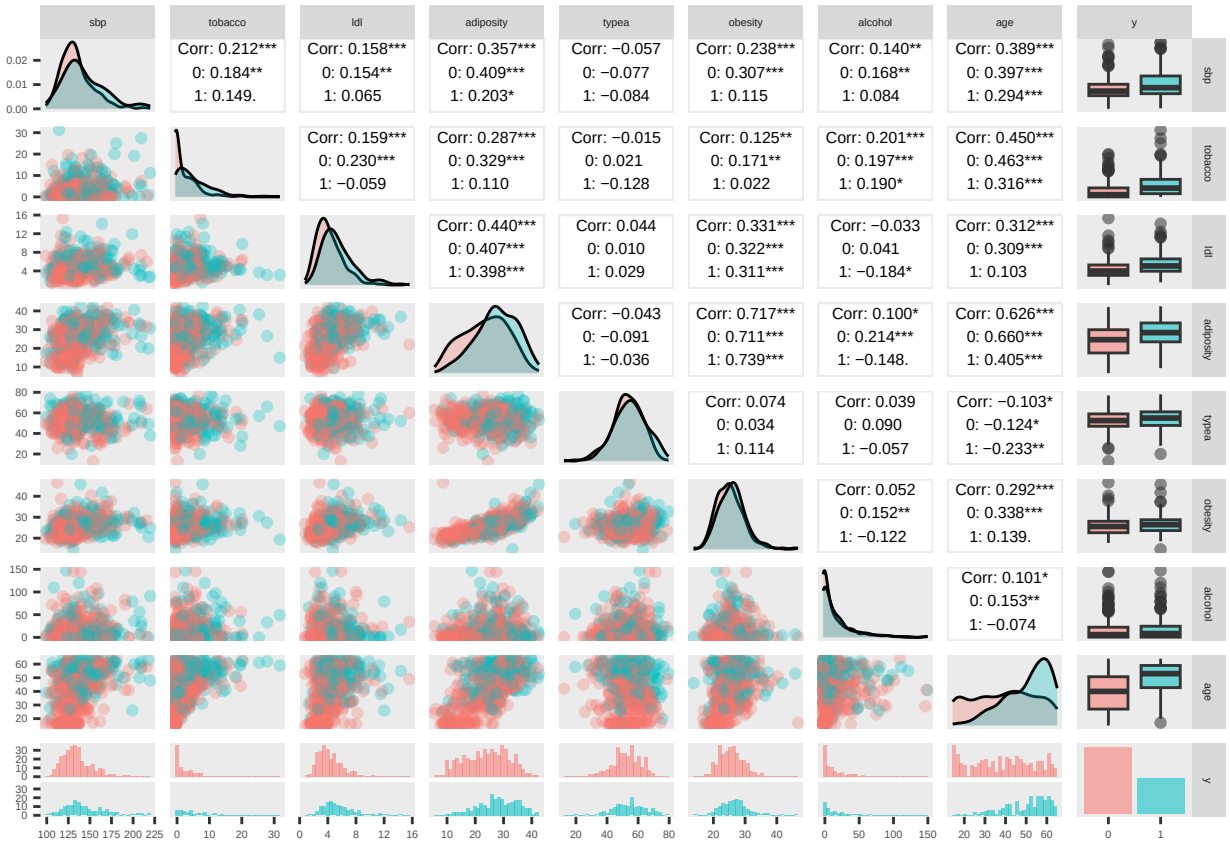


Figure 1: Distribuzione della Malattia Cardiaca

```r
# Creare lo scatter plot matrix
library(GGally)
library(ggplot2)

# Crea il pairs plot con testo più piccolo
p <- ggpairs(heart_numeric, aes(color = as.factor(y), alpha = 0.2),
             upper = list(continuous = wrap("cor", size = 2, color = "black")),
             lower = list(continuous = wrap("points", alpha = 0.3)),
```

```
            diag = list(continuous = wrap("densityDiag", alpha = 0.3))) +
  theme(panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text = element_text(size = 4),
        strip.text = element_text(size = 4))

# Mostra il grafico
print(p)
```



L'analisi esplorativa del dataset ha rivelato una distribuzione eterogenea delle variabili, con evidenti variazioni nei livelli di fattori di rischio cardiovascolare quali la pressione arteriosa sistolica (sbp), il colesterolo LDL e il consumo cumulativo di tabacco. L'esame preliminare mediante istogrammi, boxplot e matrice di correlazione ha permesso di individuare relazioni significative, in particolare tra le variabili legate allo stato cardiovascolare e i fattori demografici (come l'età), suggerendo la presenza di pattern utili per la segmentazione dei pazienti.

Le variabili più collegate tra loro (correlazione elevata) sono: - Adiposità e Obesità (0.72) → Le persone con maggiore adiposità tendono ad avere un indice di obesità più alto.

- Adiposità e Età (0.63) → L'adiposità aumenta con l'età, indicando che l'accumulo di massa grassa è correlato all'invecchiamento.
- LDL e Adiposità (0.44) → Il colesterolo LDL (lipoproteine a bassa densità) tende ad aumentare con una maggiore adiposità.
- Età e Pressione Sanguigna Sistolica (0.39) → L'aumento dell'età è associato a un incremento della pressione sanguigna.

Le correlazioni più alte indicano potenziale ridondanza tra variabili (es. obesity e adiposity e adiposity e age sono molto correlate).

Buona separazione visiva tra i cluster in alcune combinazioni, come:

- Età (age) vs. Obesità (obesity) → Differenza evidente tra gruppi
- LDL (ldl) vs. Adiposità (adiposity) → Si intravede una divisione
- Pressione (sbp) vs. Età (age) → Un po' di separazione, ma meno chiara

Molta sovrapposizione in altre combinazioni, ad esempio:

- Tipo A (typea) vs. qualsiasi variabile → Nessuna chiara separazione
- Alcol (alcohol) ha una distribuzione più dispersa, senza una chiara relazione con altre variabili

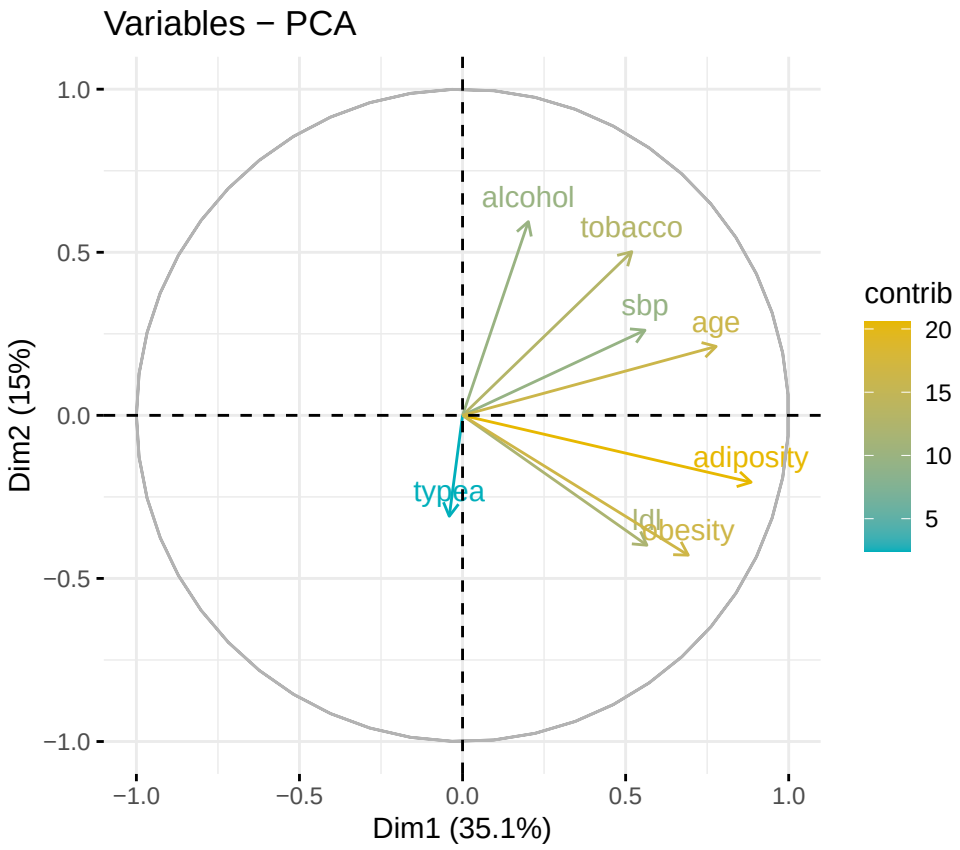
Le variabili ldl, age, adiposity, sbp sembrano avere una differenza più marcata tra i gruppi, suggerendo che potrebbero essere rilevanti nella predizione della malattia.

```
library(FactoMineR)
library(factoextra)
library(corrplot)
library(ggplot2)
library(tidyr)
library(dplyr)

mydata <- heart[, c(numeric_vars, "famhist")]

mydata$famhist <- as.factor(mydata$famhist)

res.pca <- PCA(mydata,             # ad es. con TUTTE le colonne (incl. famhist)
               scale.unit = TRUE,
               quali.sup = ncol(mydata),  # se 'famhist' è nell'ultima colonna
               graph = F)
# La variabile qualitativa supplementare famhist non influenza il calcolo
# delle componenti principali, ma viene proiettate nello spazio delle
# componenti principali per visualizzarne l'associazione con le altre variabili.

print(res.pca)
```

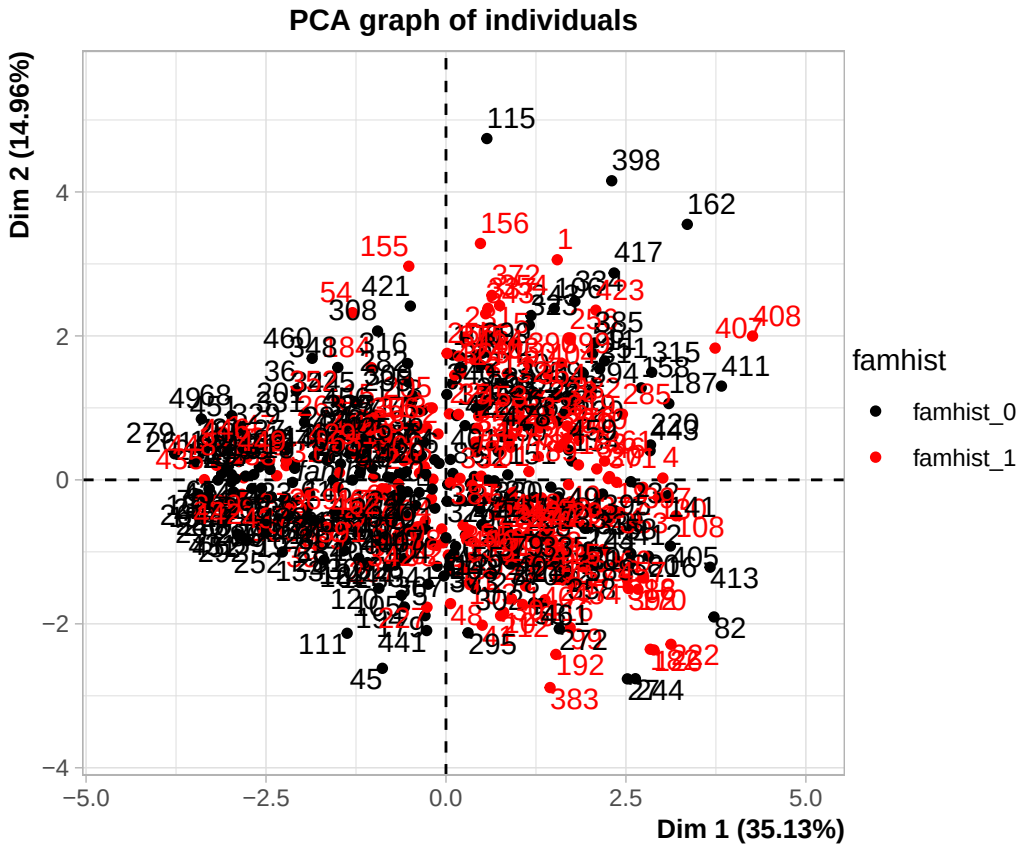
```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 462 individuals, described by 9 variables
## *The results are available in the following objects:
##
##    name                 description
## 1  "$eig"               "eigenvalues"
## 2  "$var"               "results for the variables"
## 3  "$var$coord"         "coord. for the variables"
## 4  "$var$cor"           "correlations variables - dimensions"
## 5  "$var$cos2"          "cos2 for the variables"
## 6  "$var$contrib"       "contributions of the variables"
## 7  "$ind"               "results for the individuals"
## 8  "$ind$coord"         "coord. for the individuals"
## 9  "$ind$cos2"          "cos2 for the individuals"
## 10 "$ind$contrib"       "contributions of the individuals"
## 11 "$quali.sup"         "results for the supplementary categorical variables"
## 12 "$quali.sup$coord"   "coord. for the supplementary categories"
## 13 "$quali.sup$v.test"  "v-test of the supplementary categories"
## 14 "$call"              "summary statistics"
## 15 "$call$centre"       "mean of the variables"
```

```
## 16 "$call$ecart.type"  "standard error of the variables"
## 17 "$call$row.w"       "weights for the individuals"
## 18 "$call$col.w"       "weights for the variables"
```

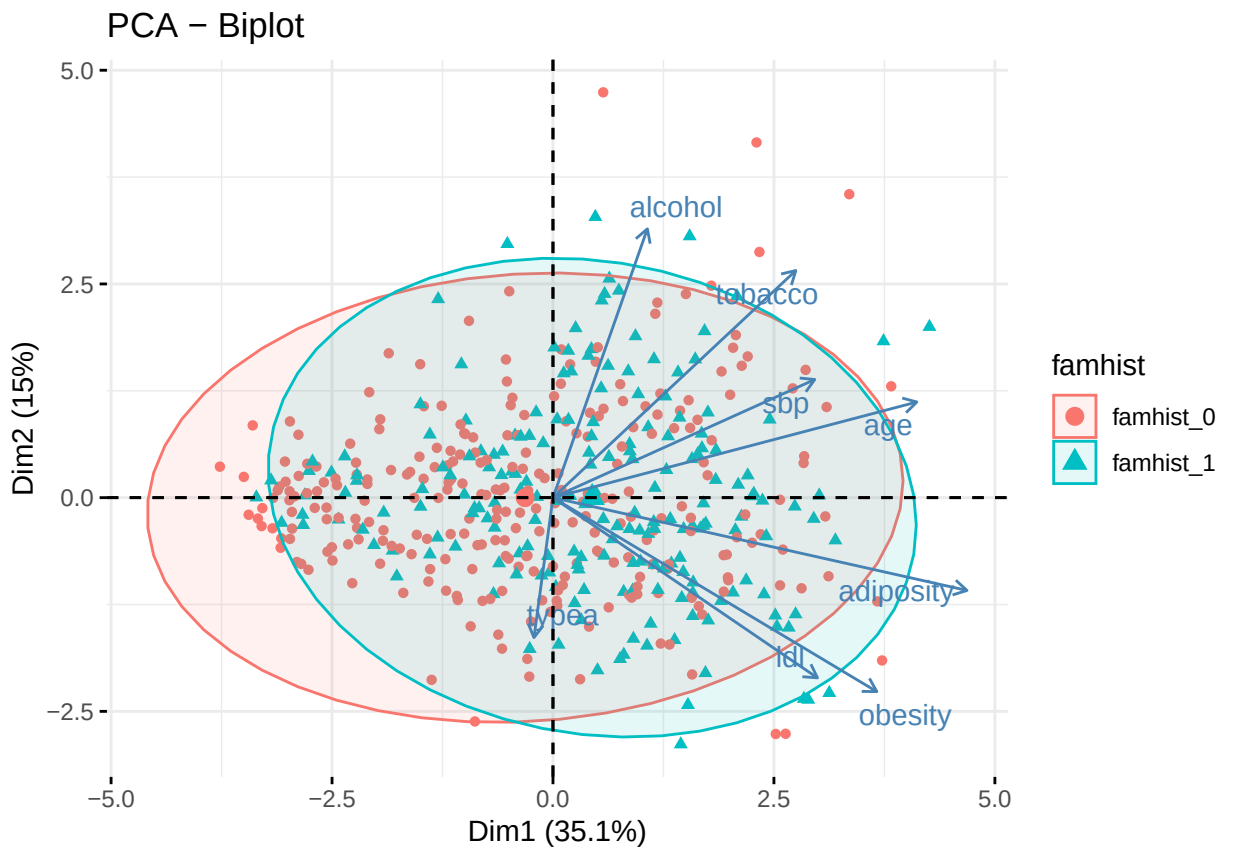
```
fviz_pca_var(res.pca, col.var = "contrib",
gradient.cols = c("#00AFBB", "#E7B800"))
```




```
plot.PCA(res.pca,
         axes=c(1,2),
         choix="ind",
         habillage = length(mydata))
```




```
fviz_pca_biplot(res.pca, label = "var", habillage = "famhist",
                addEllipses = TRUE, ellipse.level = 0.95,
                palette = c("red", "blue"), repel = TRUE)
```



La proiezione della variabile famhist nello spazio definito dalle prime due componenti principali (Dim1: 35.1% e Dim2: 15%) ha evidenziato una parziale separazione tra individui con (famhist_1) e senza (famhist_0) storia familiare di malattie cardiovascolari. I risultati mostrano che gli individui con storia familiare tendono a essere distribuiti in modo più ampio nello spazio della PCA e presentano una maggiore associazione con fattori di rischio noti per le malattie cardiovascolari, come elevati livelli di colesterolo LDL, adiposità e pressione arteriosa. Tuttavia, la sovrapposizione tra i gruppi suggerisce che famhist da solo non è un fattore discriminante sufficiente, indicando la necessità di considerare altre variabili per caratterizzare meglio il rischio cardiovascolare. Questi risultati sottolineano l'importanza dell'ereditarietà nelle malattie cardiache, ma evidenziano anche il ruolo significativo di altri fattori metabolici e comportamentali.

```r
# 1) Estraiamo la tabella dei contributi per variabile
var_contrib <- as.data.frame(res.pca$var$contrib)

# 2) Se vogliamo solo PC1 e PC2, limitiamo alle prime 2 colonne
var_contrib <- var_contrib[, 1:2]

# 3) Creiamo una colonna "Variable" con i nomi di riga
var_contrib$Variable <- rownames(var_contrib)

# 4) Ora "srotoliamo" (pivot) in formato lungo
library(tidyr)

var_contrib_long <- var_contrib %>%
  pivot_longer(
    cols = starts_with("Dim"),          # cioè "Dim.1", "Dim.2", ecc.
    names_to = "Dimension",             # la nuova colonna che conterrà "Dim.1","Dim.2",...
    values_to = "Contributo"            # la colonna che conterrà i valori numerici
```

```
  )

# 5) (facoltativo) Rinomina "Dim.1" in "PC1" e "Dim.2" in "PC2"
var_contrib_long$Dimension <- factor(
  var_contrib_long$Dimension,
  levels = c("Dim.1", "Dim.2"),
  labels = c("PC1", "PC2")
)
```

```
ggplot(var_contrib_long,
     aes(x = reorder(Variable, -Contributo),  # ordina le variabili
         y = Contributo,
         fill = Dimension)) +
geom_bar(stat = "identity", position = "stack") +
# Position = "stack" sovrappone (PC1 + PC2) in un'unica barra cumulativa

geom_text(aes(label = paste0(round(Contributo, 1), "%")),
          position = position_stack(vjust = 0.5),
          size = 3) +
# Aggiunge le etichette con i valori di contributo (in %).
# vjust = 0.5 per centrarle verticalmente nella porzione della barra.

labs(title = "Contributo delle Variabili alle Prime Due Componenti",
     x = "Variabili",
     y = "Contributo (%)") +

scale_fill_manual(values = c("PC1" = "#2E9FDF", # colore per PC1
                             "PC2" = "#E7B800")) +
# Colori personalizzati per distinguere PC1 e PC2

theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Il grafico mostra il contributo percentuale delle variabili alle prime due componenti principali (PC1 e PC2) dell'analisi delle componenti principali (PCA). La PC1 (in blu) è fortemente influenzata da obesità (17%), adiposità (27.9%) e age (21.5%), indicando che questa dimensione è legata a fattori metabolici e allo stile di vita. La PC2 (in giallo), invece, ha un contributo elevato da alcool (29.5%), tobacco (21%) e LDL (13.2%), suggerendo che questa dimensione cattura prevalentemente l'effetto del consumo di sostanze e dei parametri lipidici. L'età ha un contributo significativo su entrambe le componenti, ma con una maggiore influenza su PC1 (21.5%), suggerendo un legame tra invecchiamento e metabolismo corporeo. Complessivamente, il grafico evidenzia che la prima componente riflette principalmente aspetti metabolici e antropometrici, mentre la seconda componente rappresenta abitudini di consumo e fattori di rischio cardiovascolare. Questa distinzione è utile per comprendere la variabilità nei dati e segmentare i soggetti in gruppi con caratteristiche di rischio simili.

```
# confrontare sbp (pressione) e obesity in base a famhist
ggplot(heart, aes(x = as.factor(famhist), y = sbp, fill = as.factor(famhist))) +
  geom_boxplot(alpha=0.7) +
  labs(x="Famhist (0=Assente, 1=Presente)", y="Pressione Sistolica") +
  theme_minimal()
```

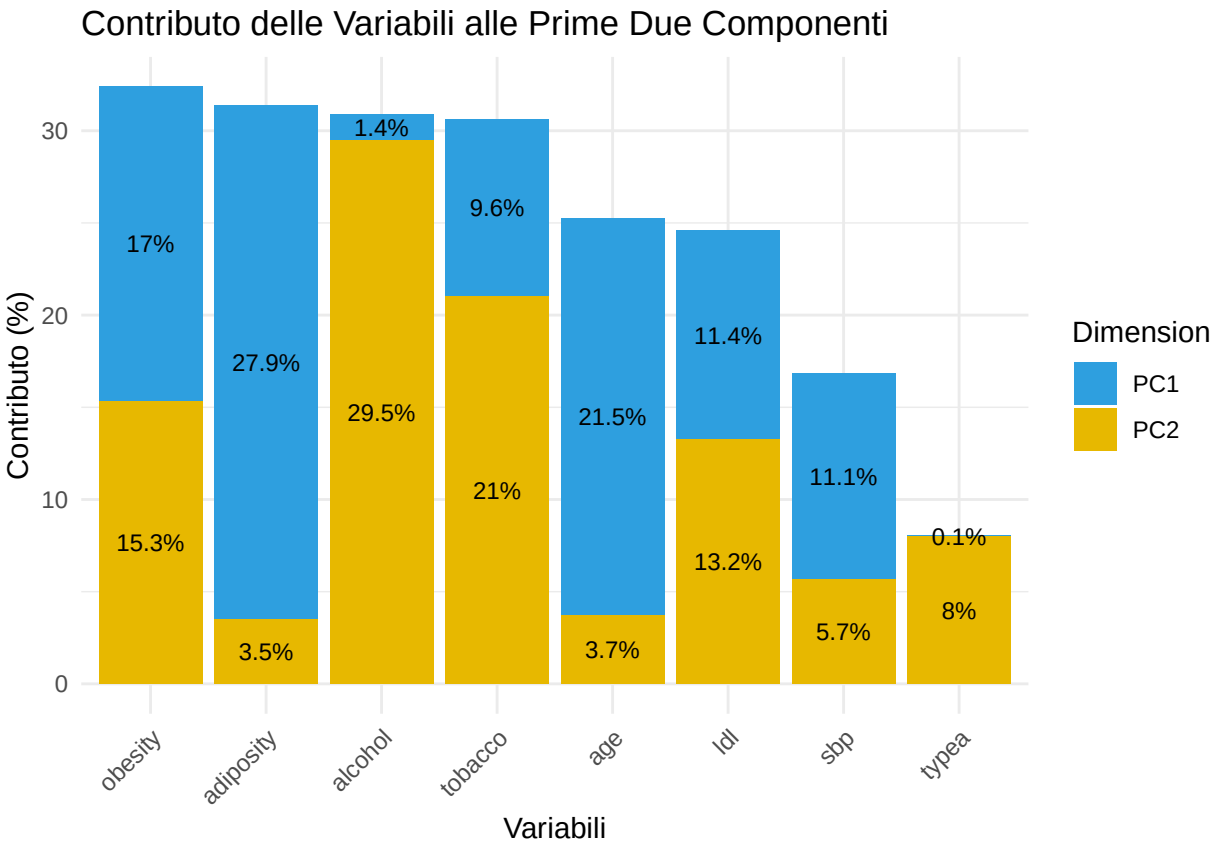
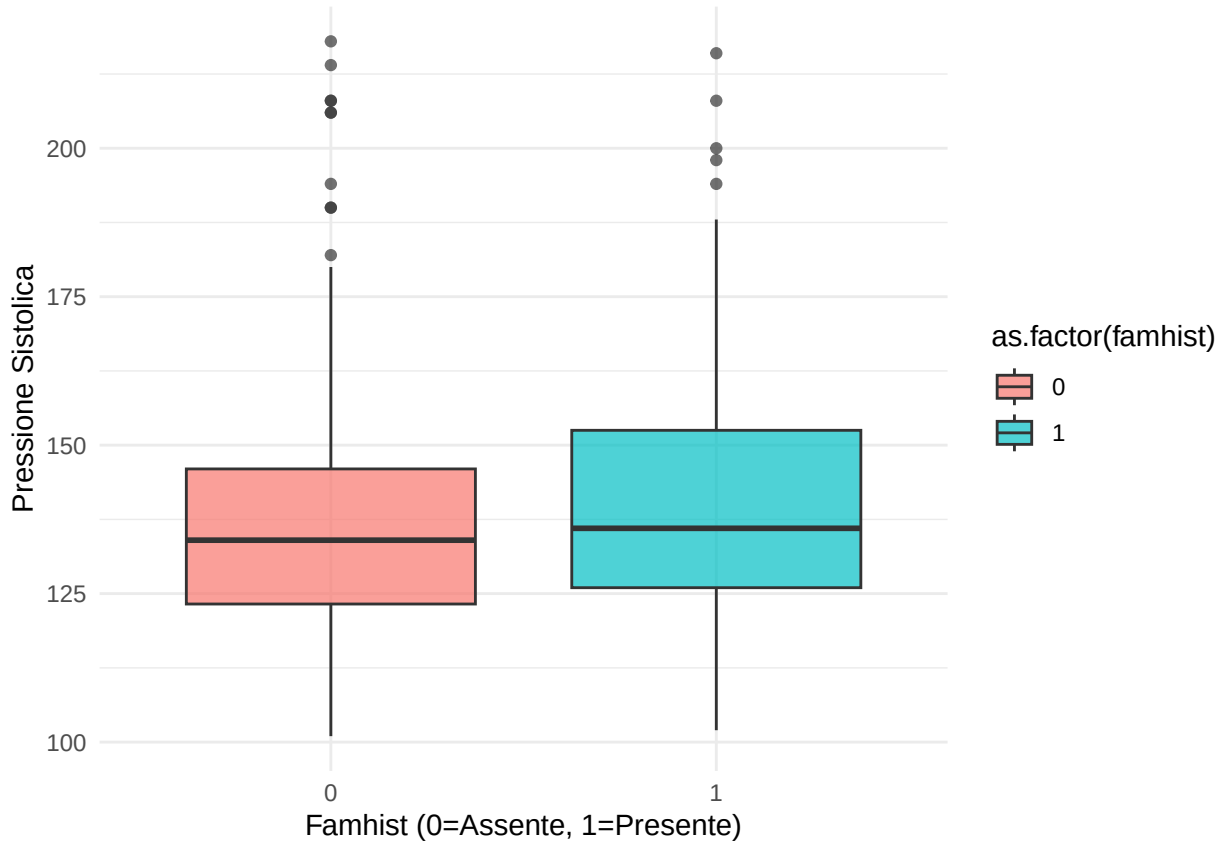
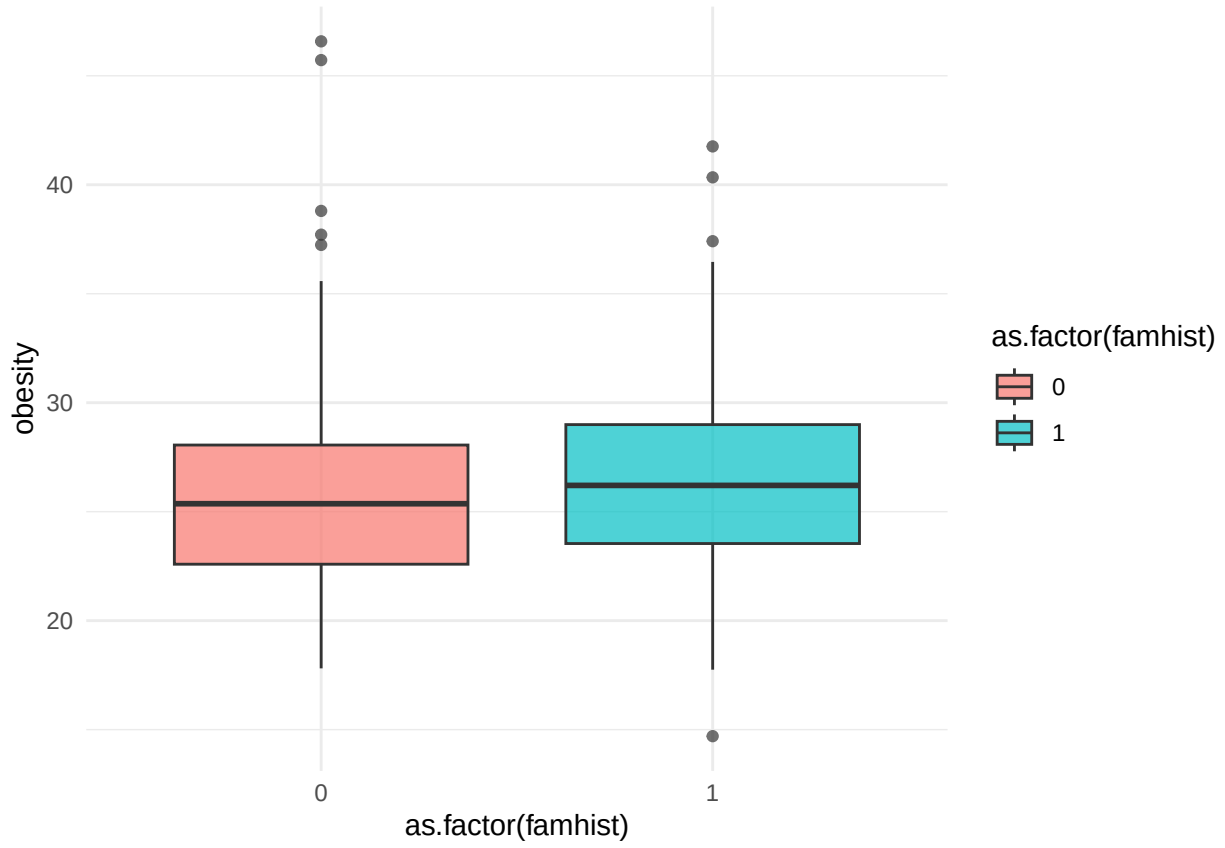


Figure 2: Visualizzazione del contributo delle variabili a PC1 e PC2




```
# Idem per obesity
ggplot(heart, aes(x = as.factor(famhist), y = obesity, fill=as.factor(famhist))) +
  geom_boxplot(alpha=0.7) +
  theme_minimal()
```




```
with(heart, t.test(sbp ~ famhist))
```

```
##
##  Welch Two Sample t-test
##
## data:  sbp by famhist
## t = -1.8487, df = 415.49, p-value = 0.0652
## alternative hypothesis: true difference in means between group 0 and group 1 is not equal to 0
## 95 percent confidence interval:
##  -7.3412759  0.2250722
## sample estimates:
## mean in group 0 mean in group 1
##        136.8481        140.4062
```

```
with(heart, wilcox.test(obesity ~ famhist))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  obesity by famhist
## W = 22028, p-value = 0.00593
## alternative hypothesis: true location shift is not equal to 0
```

```
famhist_tab <- table(heart$famhist, heart$y)
famhist_tab
```

```
##
##       0   1
```

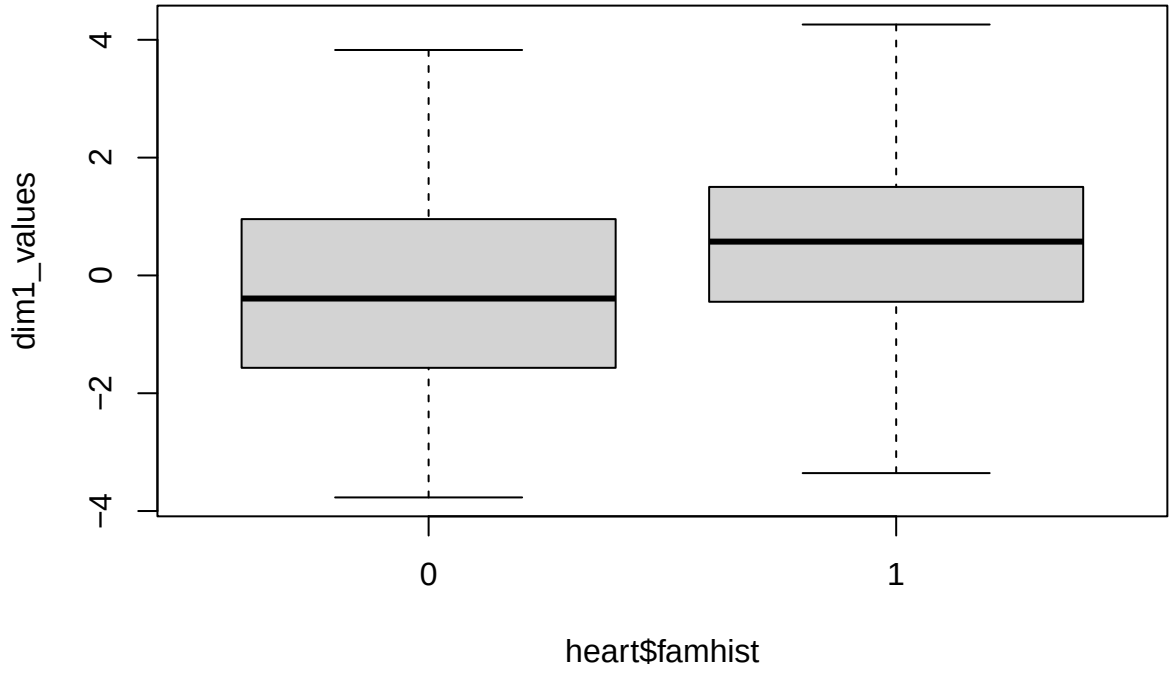
```
##   0 206  64
##   1  96  96
```

```
chisq.test(famhist_tab)
```

```
##
##  Pearson's Chi-squared test with Yates' continuity correction
##
## data:  famhist_tab
## X-squared = 33.123, df = 1, p-value = 8.653e-09
```

Boxplot di sbp vs famhist: visivamente, i soggetti con famhist=1 hanno in media una pressione sistolica leggermente più alta di quelli con famhist=0, ma non in modo estremamente marcato. Il T test su sbp da un p value = 0.0652. Non raggiunge la significatività (soglia 0.05), quindi non c'è evidenza statistica forte per dire che chi ha storia familiare abbia una pressione sistolica media diversa (più alta o più bassa). Boxplot di obesity vs famhist: i boxplot suggeriscono che i soggetti con famhist=1 hanno una distribuzione di obesità più elevata (mediana più alta). Wilcoxon test su obesity: p value = 0.006. Qui invece è significativo: c'è una differenza statisticamente rilevante tra i gruppi, indicando che chi ha storia familiare tende ad avere valori di obesità più alti. Chi-square test: p value = 8.65x10-9. Molto significativo, quindi esiste un'associazione forte tra storia familiare e presenza di malattia coronarica (y=1). In parole semplici, famhist=1 compare più spesso nei soggetti affetti dalla malattia rispetto a quelli sani.

```
dim1_values <- res.pca$ind$coord[,1]
boxplot(dim1_values ~ heart$famhist)
```



Dato che PC1 (in quest'analisi) risulta fortemente associata a obesità/adiposità/LDL/età (a seconda dei contributi), osservare un valore più elevato per famhist=1 suggerisce che chi ha storia familiare tende ad avere punteggi più alti su quell'asse metabolico

```
# Utilizziamo le variabili obesity, ldl, age, sbp tra quelle che sembrano avere
# una differenza più marcata tra i gruppi e anche un alto contributo
# nelle due componenti principali (PCA)

set.seed(123)
vars <- c("obesity","ldl", "age","sbp")
heart_vars=data.frame(heart[, vars])

library(hopkins)
print(hopkins(heart_vars))
```

```
## [1] 0.9658282
```

```
# il dataset è clasterizzabile se il valore di Hopkins è vicino a 1

# Creare la matrice di scatterplots con statistiche
ggpairs(heart[, vars],
        aes(color = as.factor(heart$y), alpha = 0.4)) +
  theme_minimal()
```

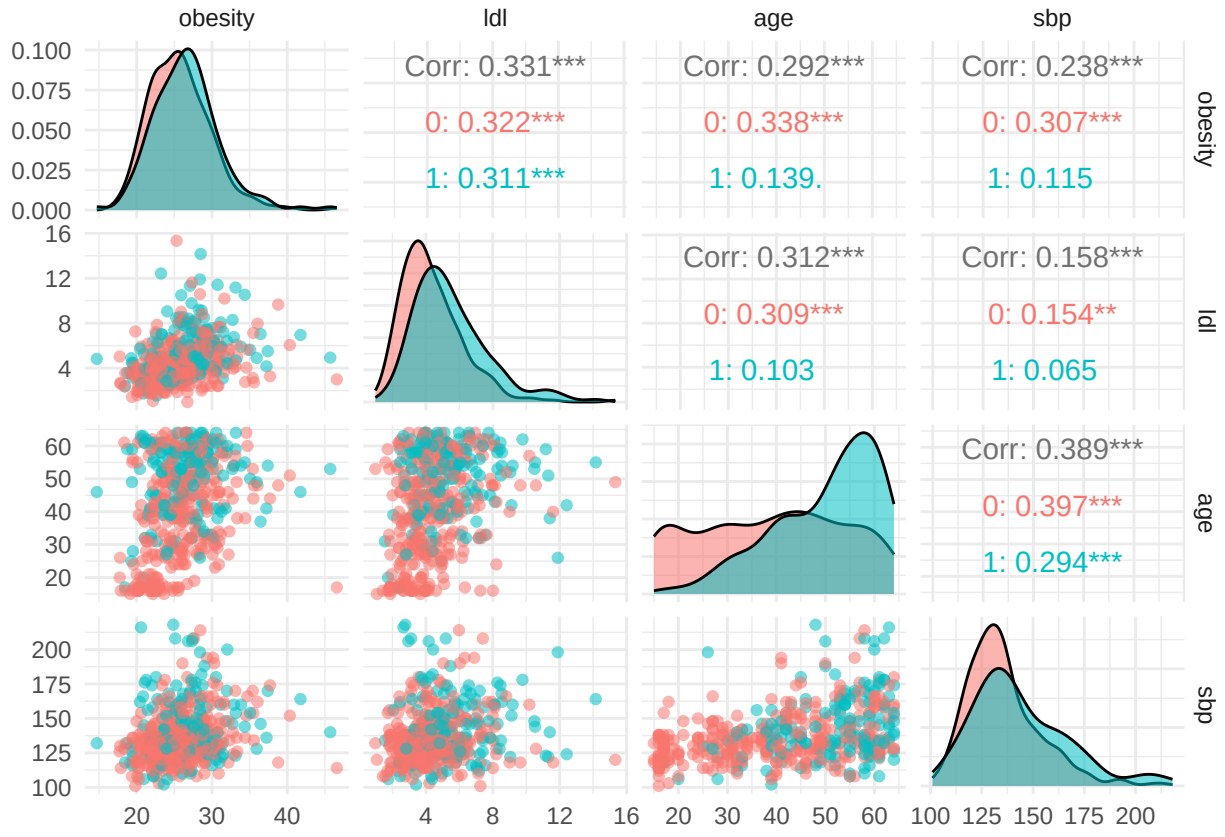


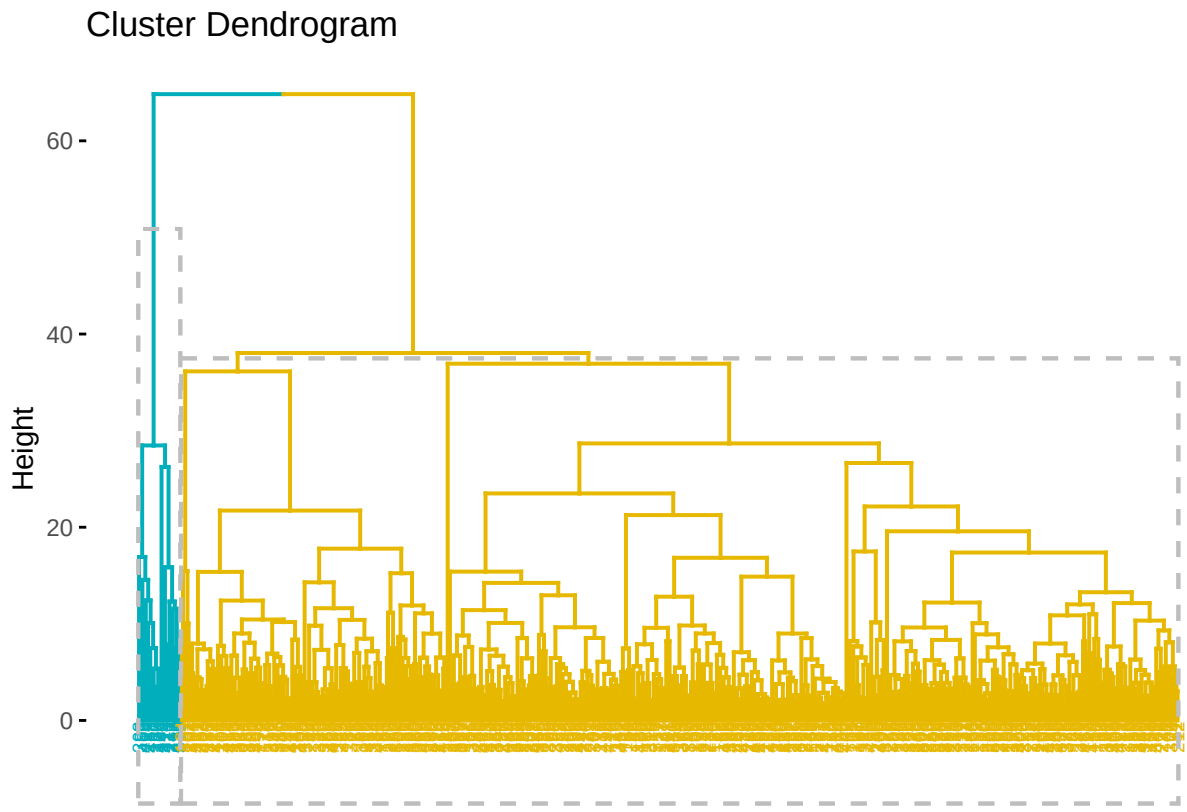
Figure 3: Matrice di scatterplots

```
# Analisi dei possibili metodi di clustering:
#AGglomerative NESting (Hierarchical Clustering)

# Caricamento delle librerie necessarie
```

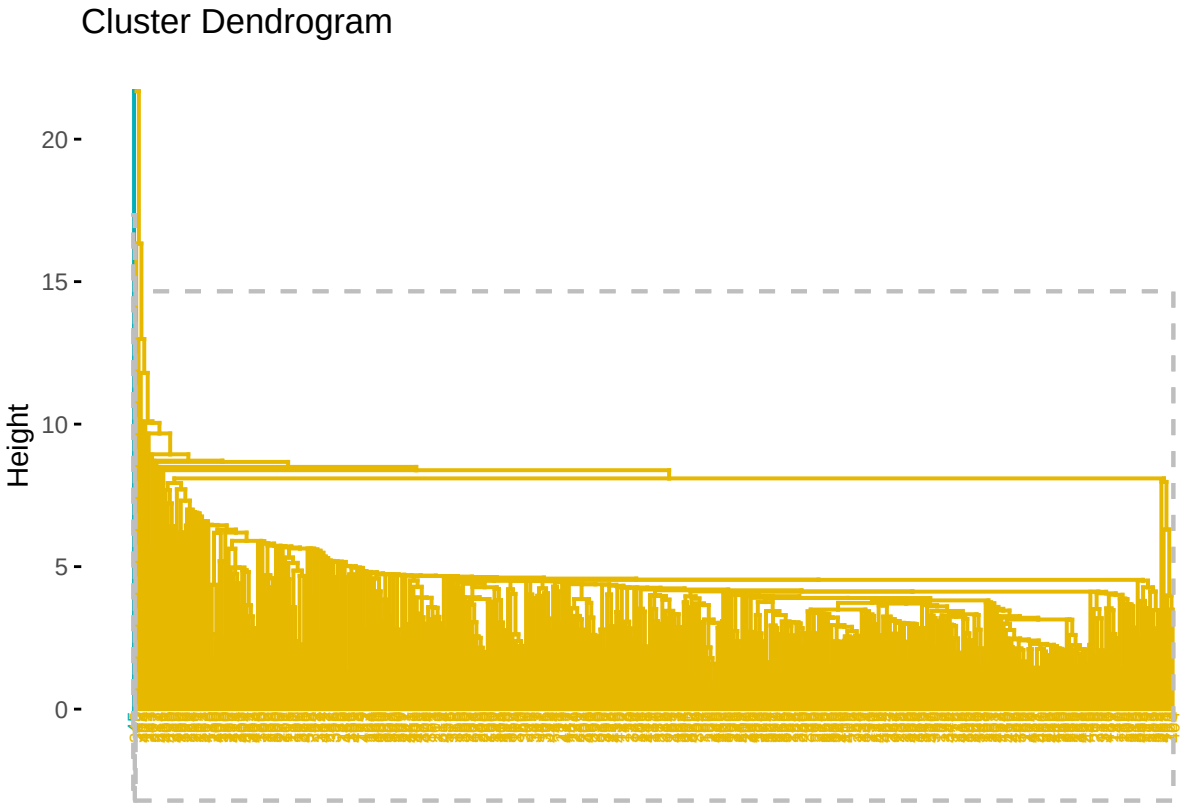
```r
library(cluster)
library(factoextra)
library(clValid)

# Definizione dei metodi di linkage
linkage_methods <- c("average", "single", "complete", "ward.D2", "centroid")

# Creazione di una lista per salvare i risultati
hclust_results <- list()
silhouette_scores <- numeric(length(linkage_methods))
dunn_scores <- numeric(length(linkage_methods))
connectivity_scores <- numeric(length(linkage_methods))

# Calcolo della matrice di distanza
if (any(sapply(heart_vars, is.factor))) {
  dist_matrix <- daisy(heart_vars, metric = "gower")
} else {
  dist_matrix <- dist(heart_vars, method = "euclidean")
}

# Loop per eseguire hclust con diversi metodi di linkage
for (i in seq_along(linkage_methods)) {
  method <- linkage_methods[i]
  cat("\n### Metodo:", method, "\n")
  cat("Eseguendo hclust con metodo:", method, "\n")

  # Esegui hclust
  hclust_results[[method]] <- hclust(dist_matrix, method = method)

  # Creazione dei cluster (impostiamo k = 2 per confronto)
  clusters <- cutree(hclust_results[[method]], k = 2)

  # Visualizzazione del dendrogramma
  plot_dend <- fviz_dend(hclust_results[[method]], k = 2,
                         cex = 0.5,
                         k_colors = c("#00AFBB", "#E7B800"),
                         color_labels_by_k = TRUE,
                         rect = TRUE)

  print(plot_dend)

  # Calcolo Indice di Silhouette
  sil <- silhouette(clusters, dist_matrix)
  silhouette_scores[i] <- mean(sil[, 3])  # Media silhouette

  # Calcolo Indice di Dunn
  dunn_scores[i] <- dunn(dist_matrix, clusters)

  # Calcolo Indice di Connectivity
  connectivity_scores[i] <- connectivity(distance = dist_matrix,
                                         clusters = clusters)
}
```

```
##
```

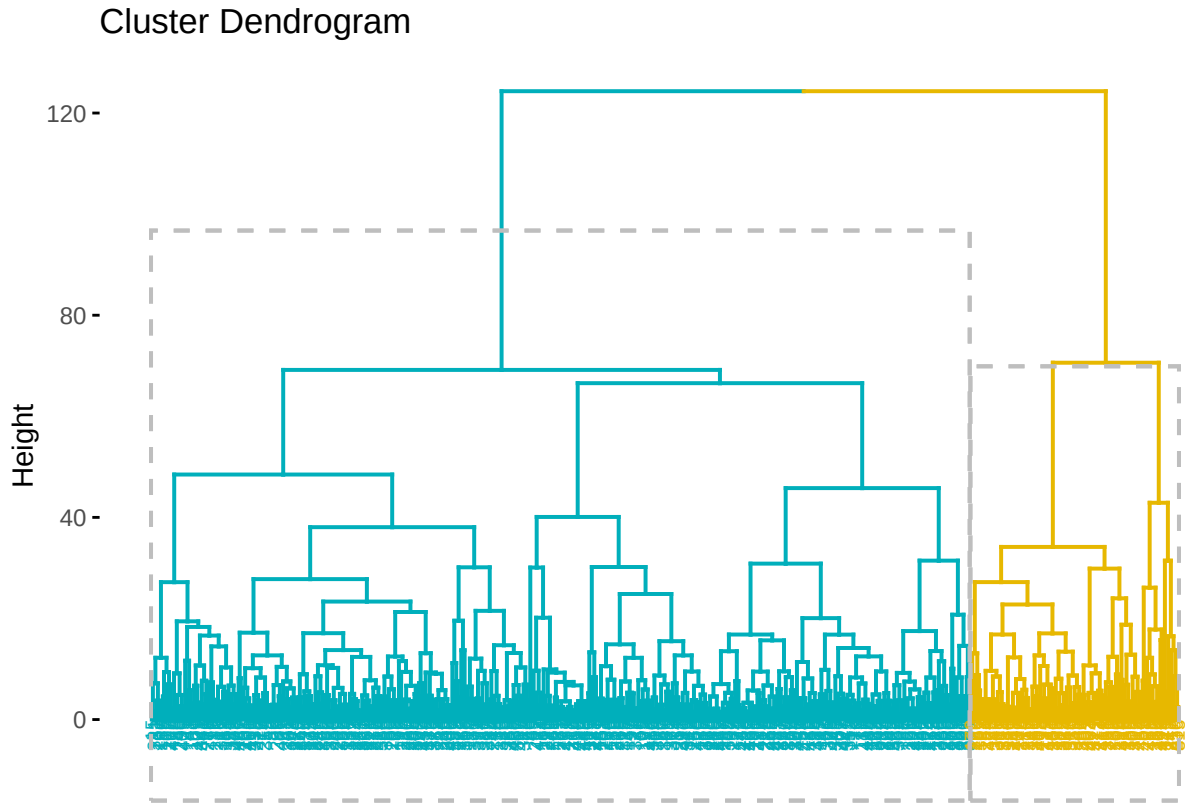
```
## ### Metodo: average
## Eseguendo hclust con metodo: average
```



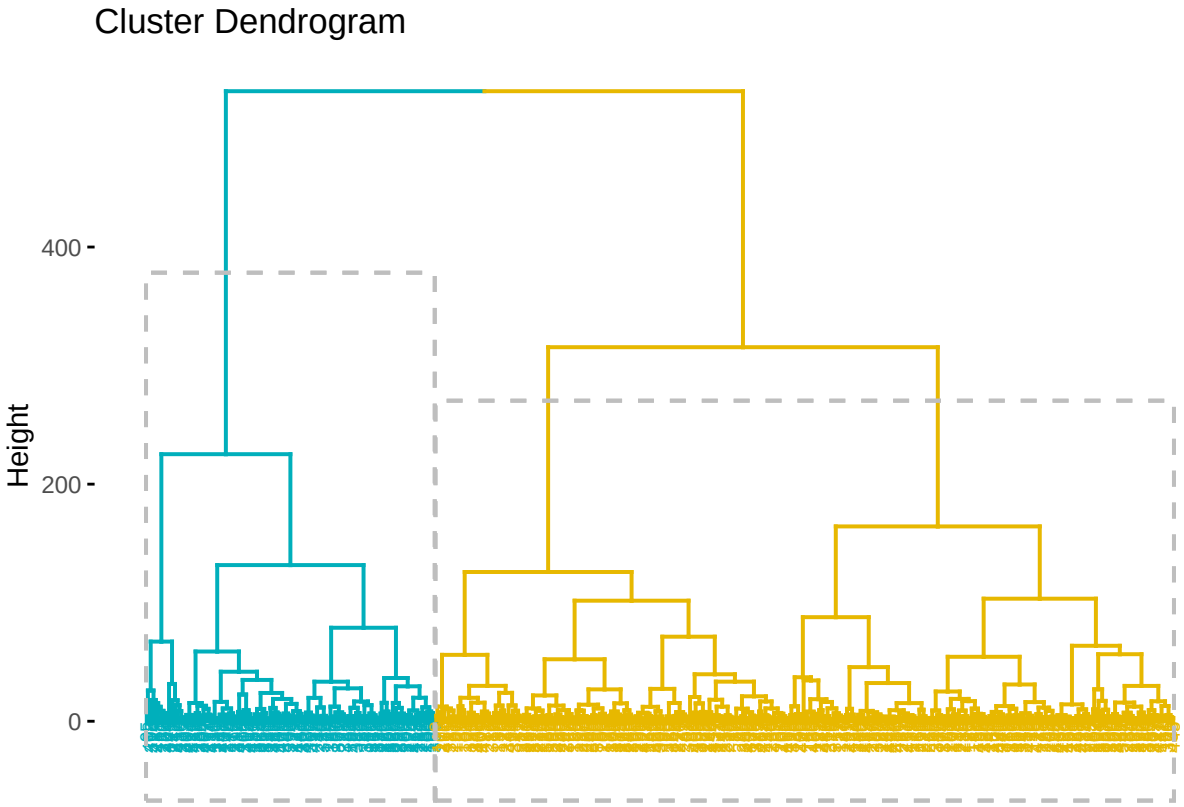

```
##
## ### Metodo: single
## Eseguendo hclust con metodo: single
```



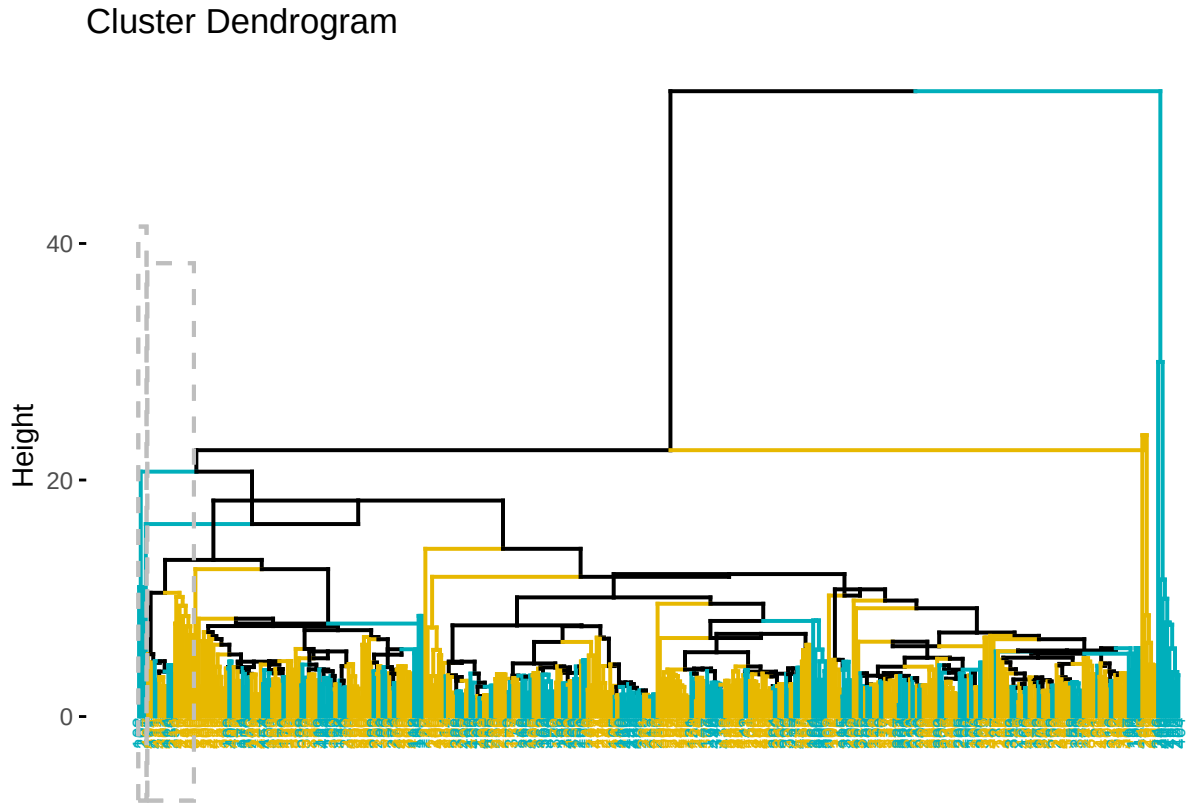

```
##
## ### Metodo: complete
## Eseguendo hclust con metodo: complete
```




```
##
## ### Metodo: ward.D2
## Eseguendo hclust con metodo: ward.D2
```




```
##
## ### Metodo: centroid
## Eseguendo hclust con metodo: centroid
```




```
# Creazione della tabella riassuntiva
validation_summary <- data.frame(
  method = linkage_methods,
  silhouette_score = silhouette_scores,
  dunn_index = dunn_scores,
  connectivity = connectivity_scores
)

library(knitr)
kable(validation_summary, caption = "Confronto tra i Metodi di Hierarchical Clustering")
```

Table 1: Confronto tra i Metodi di Hierarchical Clustering

method	silhouette_score	dunn_index	connectivity
average	0.5168602	0.0855237	6.461905
single	0.2547516	0.1744099	2.928968
complete	0.4551268	0.0544221	25.942857
ward.D2	0.4310708	0.0409879	22.196032
centroid	0.5651414	0.0755448	7.619841

Abbiamo confrontato 5 metodi di clustering gerarchico (average linkage, single linkage, complete linkage, Ward.D2 e centroid linkage) utilizzando tre metriche di valutazione: Silhouette Score, Dunn Index e Connectivity.

Il Silhouette Score, che misura la coesione interna dei cluster e la loro separabilità, è più elevato per il metodo centroid (0.5651) e per il metodo average (0.5169), suggerendo che questi metodi producono cluster più

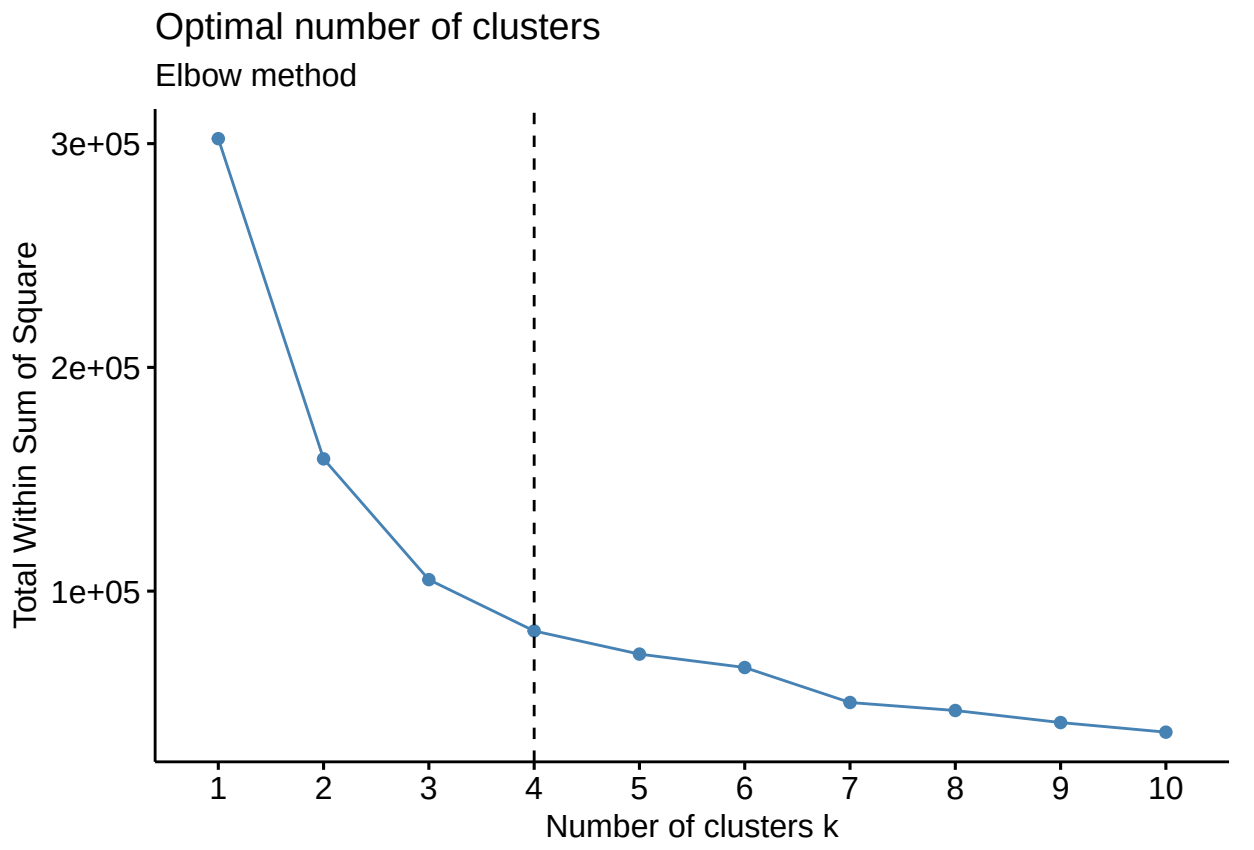
distinti e ben separati. Al contrario, il metodo single linkage ha ottenuto il punteggio più basso (0.2548), indicando una scarsa separazione tra i cluster.

L'indice di Dunn, che valuta la compattezza e la separazione tra cluster, è più alto per il metodo single linkage (0.1744), ma questo metodo è noto per la tendenza a formare cluster allungati e poco stabili. Al contrario, i metodi complete linkage (0.0544) e ward.D2 (0.0409) presentano valori inferiori, suggerendo cluster più compatti ma potenzialmente meno separabili.

La connectivity, che misura la stabilità della struttura dei cluster (valori più bassi indicano clustering più definito), è più bassa per il metodo single (2.9289), mentre i metodi complete (25.94) e ward.D2 (22.19) presentano valori più alti, indicando una possibile frammentazione della struttura dei cluster.

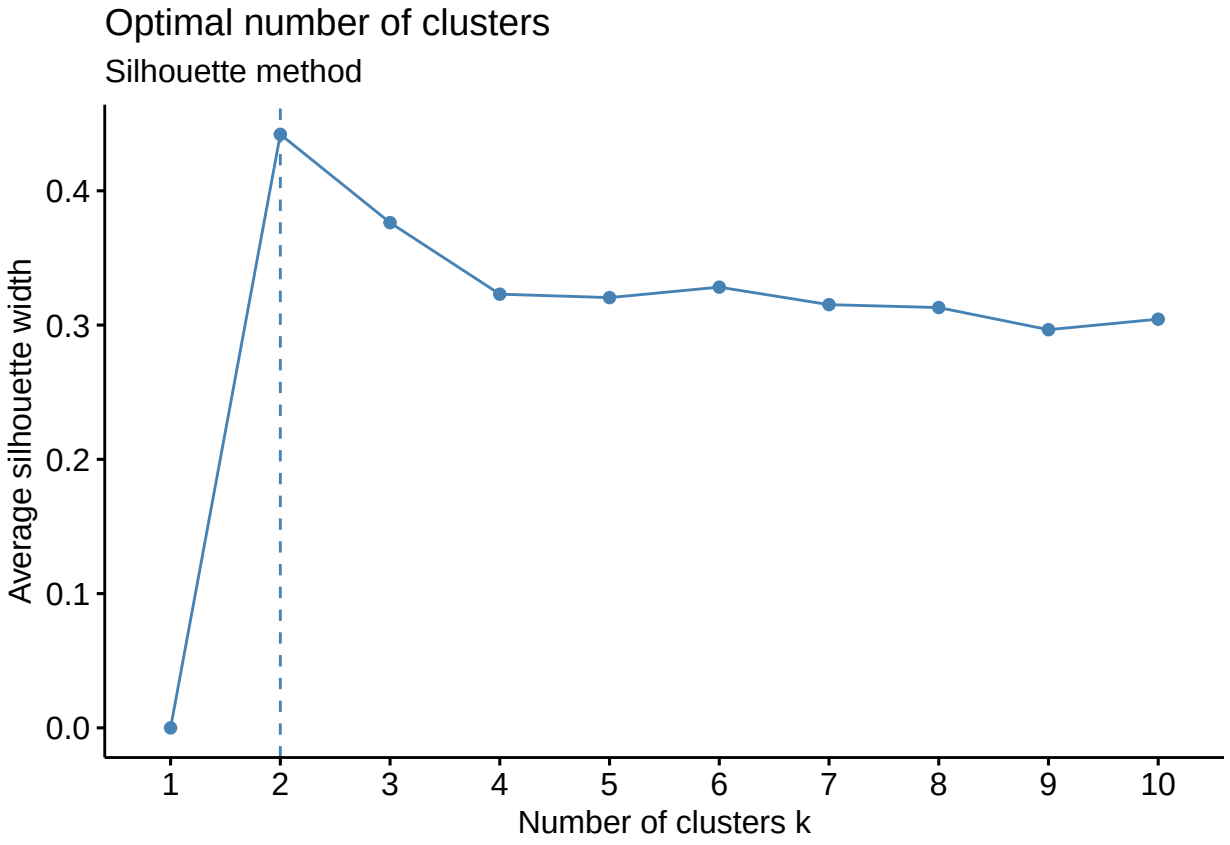
Nel complesso, il metodo centroid sembra essere il più adatto per questa analisi, in quanto mostra il miglior Silhouette Score e una buona separazione tra cluster, mentre il metodo average rappresenta un buon compromesso tra coesione e separabilità. Il metodo single linkage, pur avendo un buon Dunn Index, tende a creare cluster meno distinti e più allungati, rendendolo meno adatto per questa analisi.

```
library(factoextra)
library(NbClust)

df.scaled <- as.data.frame(heart_vars)

# Elbow method
fviz_nbclust(df.scaled, kmeans, method = "wss") +
  geom_vline(xintercept = 4, linetype = 2)+
  labs(subtitle = "Elbow method")
```

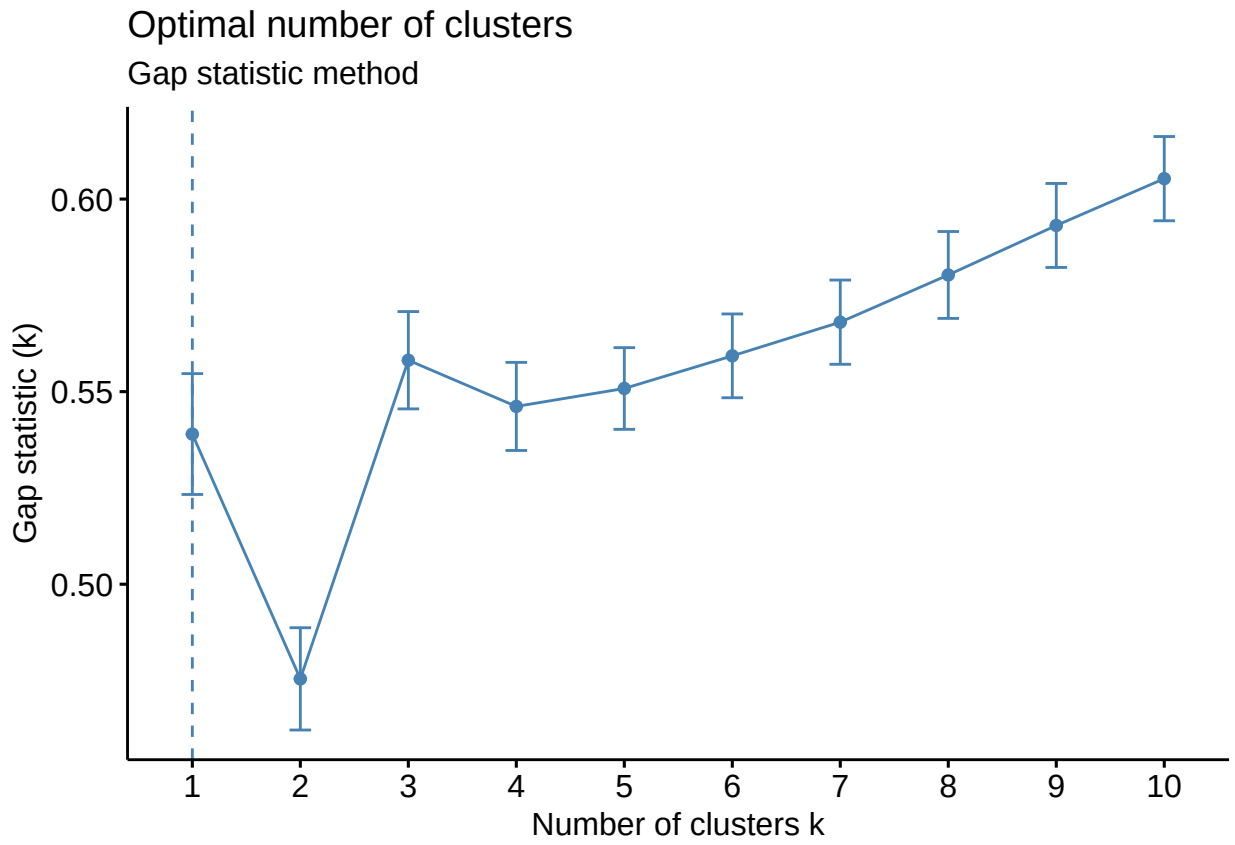



```
# Silhouette method
fviz_nbclust(df.scaled, kmeans, method = "silhouette")+
```

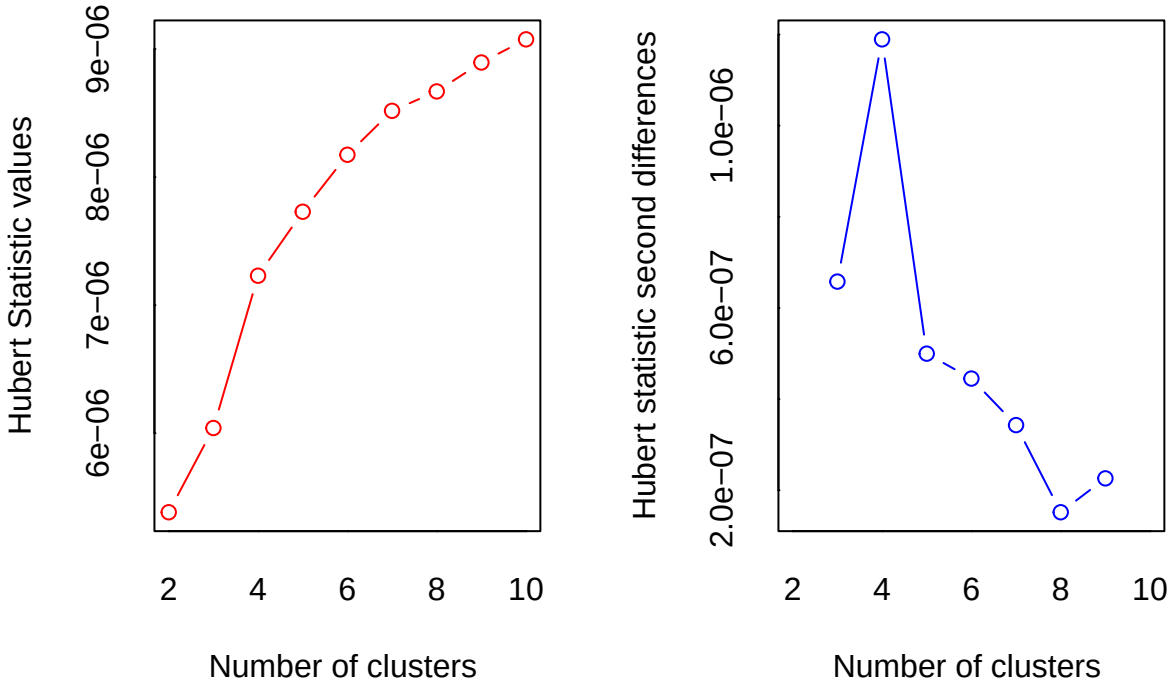
```
  labs(subtitle = "Silhouette method")
```



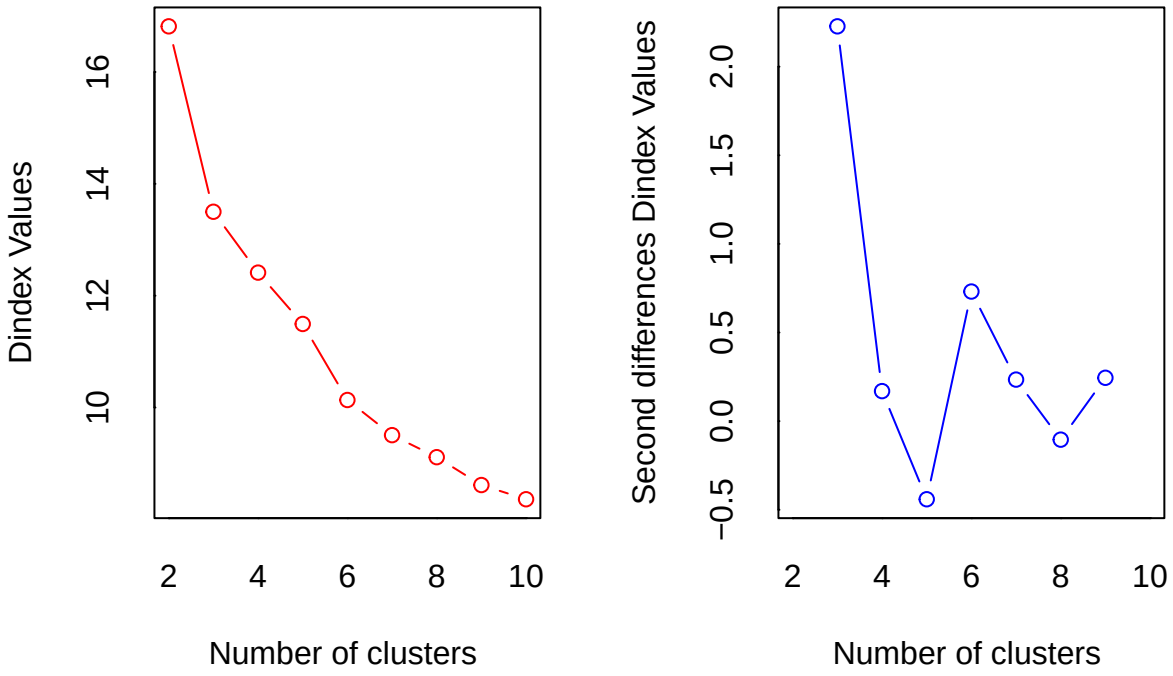

```
# Gap statistic
set.seed(123)
fviz_nbclust(df.scaled, kmeans, nstart = 25, method = "gap_stat", nboot = 500)+
  labs(subtitle = "Gap statistic method")
```




```
## NbClust function
library("NbClust")
nb <- NbClust(df.scaled, distance = "euclidean", min.nc = 2,
              max.nc = 10, method = "kmeans")
```




```
## *** : The Hubert index is a graphical method of determining the number of clusters.
##                 In the plot of Hubert index, we seek a significant knee that corresponds to a
##                 significant increase of the value of the measure i.e the significant peak in Hubert
##                 index second differences plot.
##
```




```
## *** : The D index is a graphical method of determining the number of clusters.
##                 In the plot of D index, we seek a significant knee (the significant peak in Dindex
##                 second differences plot) that corresponds to a significant increase of the value of
##                 the measure.
##
## *******************************************************************
## * Among all indices:
## * 8 proposed 2 as the best number of clusters
## * 8 proposed 3 as the best number of clusters
## * 1 proposed 4 as the best number of clusters
## * 2 proposed 6 as the best number of clusters
## * 1 proposed 7 as the best number of clusters
## * 2 proposed 8 as the best number of clusters
## * 1 proposed 9 as the best number of clusters
## * 1 proposed 10 as the best number of clusters
##
##                     ***** Conclusion *****
##
## * According to the majority rule, the best number of clusters is  2
##
##
## *******************************************************************
```

L'analisi è stata condotta utilizzando diversi metodi per determinare il numero ottimale di cluster nel dataset heart. Ecco i risultati ottenuti:

Metodo del Gomito (Elbow Method): il punto di flesso si trova a k = 4, suggerendo che quattro cluster siano

un buon compromesso tra la riduzione della varianza intra-cluster e la semplicità del modello. Metodo della Silhouette: il valore massimo dell'indice di silhouette si ottiene con k = 2, indicando che la separazione tra i cluster è più netta quando si usano due gruppi. Gap Statistic: il valore massimo della statistica di gap suggerisce k = 2 come scelta ottimale. Hubert Statistic: il picco nel secondo differenziale indica che k = 4 o k = 5 potrebbero essere scelte adeguate. D-Index: il valore significativo del secondo differenziale suggerisce che il numero ottimale di cluster sia k = 4. In conclusione, la maggioranza degli indici suggerisce che il numero ottimale di cluster sia k = 2, mentre alcuni metodi indicano k = 4 come alternativa valida.

Se l'obiettivo è massimizzare la separazione tra cluster, allora k = 2 è la scelta migliore (parsimonia). Se l'obiettivo è identificare sottogruppi più dettagliati, allora k = 4 potrebbe essere una scelta.

```
# Utilizziamo Partitioning clustering
k <- 2
# K-Means
set.seed(123)
km.res <- kmeans(df.scaled , centers = k, nstart = 25)

# K-Medoids (PAM)
set.seed(123)
pam.res <- pam(df.scaled , k, metric = "euclidean", nstart = 25)

# CLARA
set.seed(123)
clara.res <- clara(df.scaled , k, metric = "euclidean", samples = 5)

# Silhouette Score per Confronto
km.sil <- silhouette(km.res$cluster, dist(df.scaled ))
pam.sil <- silhouette(pam.res$clustering, dist(df.scaled ))
clara.sil <- silhouette(clara.res$clustering, dist(df.scaled ))

# Estrarre il valore medio di silhouette per ogni metodo
silhouette_values <- data.frame(
  Metodo = c("K-Means", "K-Medoids (PAM)", "CLARA"),
  Media_Silhouette = c(mean(km.sil[, 3]),
                       mean(pam.sil[, 3]),
                       mean(clara.sil[, 3]))
)

library(knitr)
silhouette_values<-silhouette_values %>%
  mutate(across(where(is.numeric), ~ signif(.x, 3)))
print(silhouette_values)
```

```
##            Metodo Media_Silhouette
## 1         K-Means            0.442
## 2 K-Medoids (PAM)            0.432
## 3           CLARA            0.445
```

```
kable(silhouette_values, format = "latex",   caption = "Confronto dell'Indice
      di Silhouette tra i Metodi di Clustering")
```

```
# Visualizzazione dei cluster
p1 <- fviz_cluster(km.res, data = df.scaled,
             palette = c("#00AFBB", "#FC4E07"),
             geom = "point",  # Mostra solo i punti
             pointsize = 3,   # Aumenta la dimensione dei punti
```

Table 2: Confronto dell'Indice di Silhouette tra i Metodi di Clustering

Metodo	Media_Silhouette
K-Means	0.442
K-Medoids (PAM)	0.432
CLARA	0.445

```
            alpha = 0.6,      # Aggiunge trasparenza
            ellipse.type = "convex",  # Migliora la separazione dei cluster
            star.plot = TRUE,  # Collega i punti ai centroidi
            repel = TRUE,  # Evita sovrapposizioni di etichette
            ggtheme = theme_minimal()) +
      ggtitle("K-Means")

p2 <- fviz_cluster(pam.res, data = df.scaled,
            palette = c("#00AFBB", "#FC4E07"),
            geom = "point",
            pointsize = 3,
            alpha = 0.6,
            ellipse.type = "convex",
            star.plot = TRUE,
            repel = TRUE,
            ggtheme = theme_minimal()) +
      ggtitle("K-Medoids (PAM)")

p3 <- fviz_cluster(clara.res, data = df.scaled,
            palette = c("#00AFBB", "#FC4E07"),
            geom = "point",
            pointsize = 3,
            alpha = 0.6,
            ellipse.type = "convex",
            star.plot = TRUE,
            repel = TRUE,
            ggtheme = theme_minimal()) +
      ggtitle("CLARA")


# Visualizzazione Silhouette Score
p4 <- fviz_silhouette(km.sil) + ggtitle("Silhouette - K-Means")
```

```
##   cluster size ave.sil.width
## 1       1  125          0.43
## 2       2  337          0.45
```

```
p5 <- fviz_silhouette(pam.sil) + ggtitle("Silhouette - PAM")
```

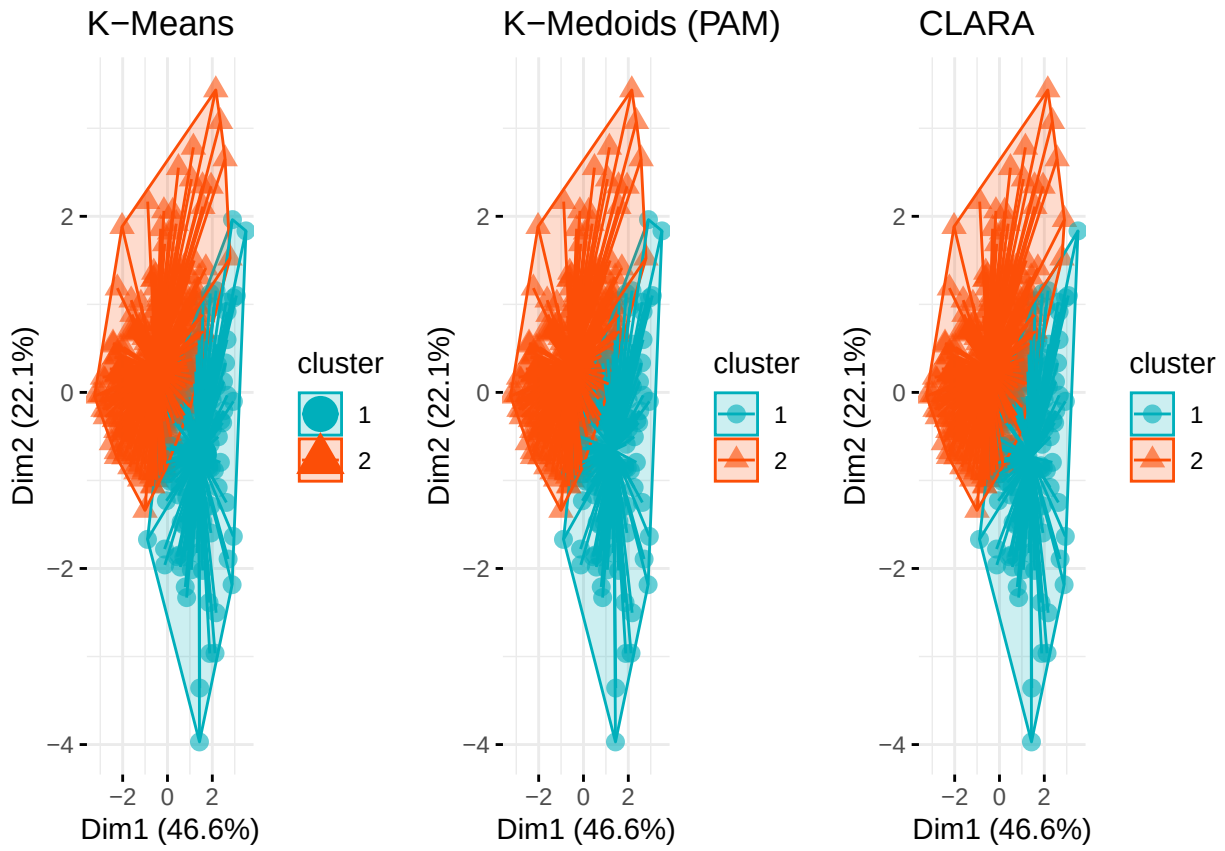
```
##   cluster size ave.sil.width
## 1       1  134          0.41
## 2       2  328          0.44
```

```
p6 <- fviz_silhouette(clara.sil) + ggtitle("Silhouette - CLARA")
```

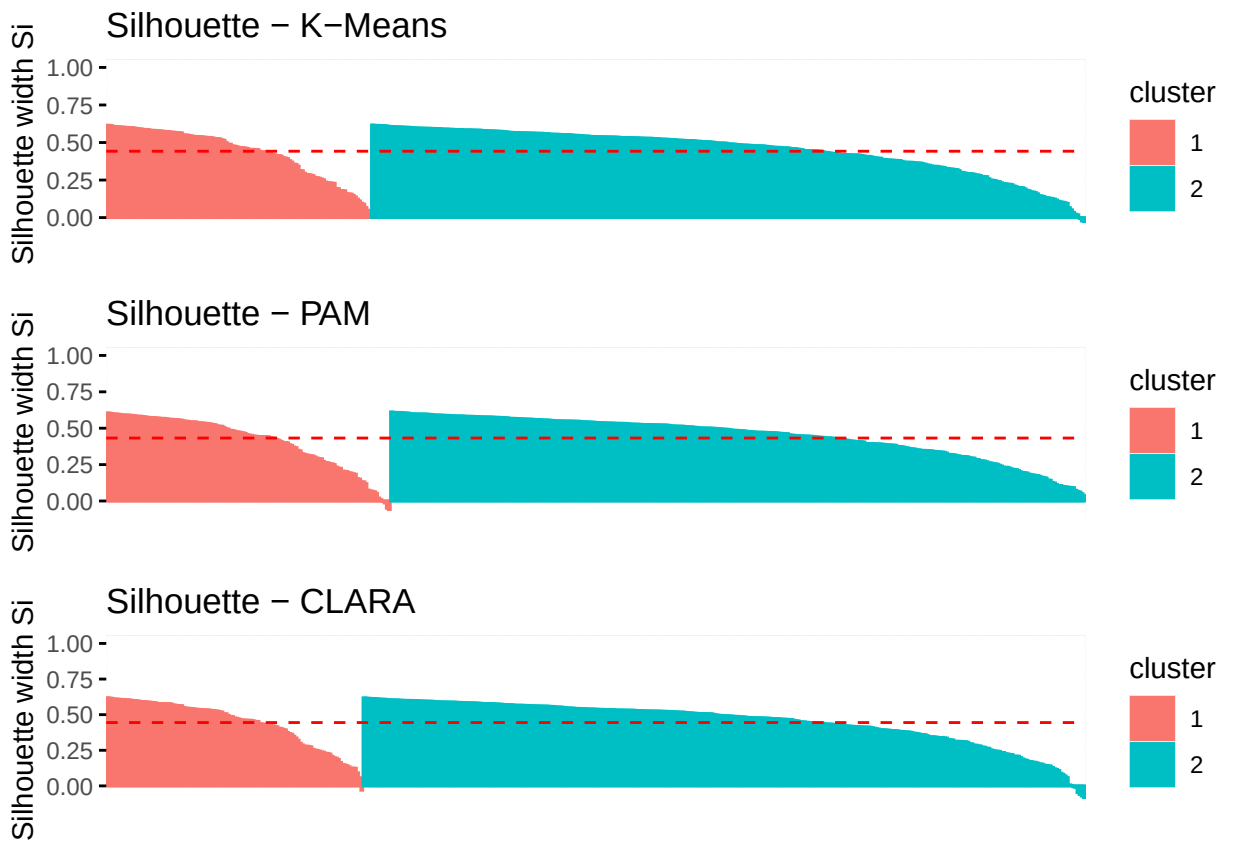
```
##   cluster size ave.sil.width
## 1       1  121          0.44
```

```
## 2       2  341          0.45
```

```
# Mostra i grafici
library(gridExtra)
grid.arrange(p1, p2, p3, ncol = 3)
```




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



Il confronto tra i metodi di clustering applicati al dataset heart ha evidenziato tre approcci principali: K-Means, K-Medoids (PAM) e CLARA. Ognuno di questi metodi ha vantaggi e limitazioni a seconda delle caratteristiche dei dati.

Il K-Means è un algoritmo rapido ed efficace che suddivide i dati minimizzando la varianza interna ai cluster. Tuttavia, assume che i cluster abbiano una forma sferica e sia di dimensioni simili, rendendolo sensibile agli outlier. Al contrario, il PAM (K-Medoids) utilizza punti reali del dataset come medoid, risultando più robusto rispetto agli outlier e adatto a distribuzioni non gaussiane. CLARA, invece, è una variante ottimizzata di PAM progettata per dataset di grandi dimensioni, che campiona i dati per migliorare l'efficienza computazionale.

Dai grafici che rappresentano i cluster su due dimensioni principali, è evidente una separazione chiara tra due gruppi distinti, il che conferma che due cluster siano una scelta adeguata. Inoltre, il confronto degli indici di silhouette, che misurano la qualità della separazione tra i cluster, mostra che CLARA ottiene il valore più alto (0.445), seguito da K-Means (0.442) e PAM (0.432). Questo significa che CLARA fornisce la migliore separazione tra cluster, sebbene la differenza con K-Means sia minima.

In termini di scelta del metodo migliore, CLARA è l'opzione più indicata per dataset di grandi dimensioni, mentre K-Means è un'alternativa veloce con risultati simili. PAM, invece, potrebbe essere preferibile in presenza di outlier o distribuzioni non gaussiane, ma ha ottenuto un valore di silhouette leggermente inferiore.

```
# Aggiunge al dataset heart i cluster trovati da CLARA
heart_clustered <- heart
heart_clustered$cluster <- factor(clara.res$clustering)  # Assegna i cluster di CLARA

library(dplyr)
summary_by_cluster <- heart_clustered %>%
  group_by(cluster) %>%
  summarise(across(where(is.numeric),
```

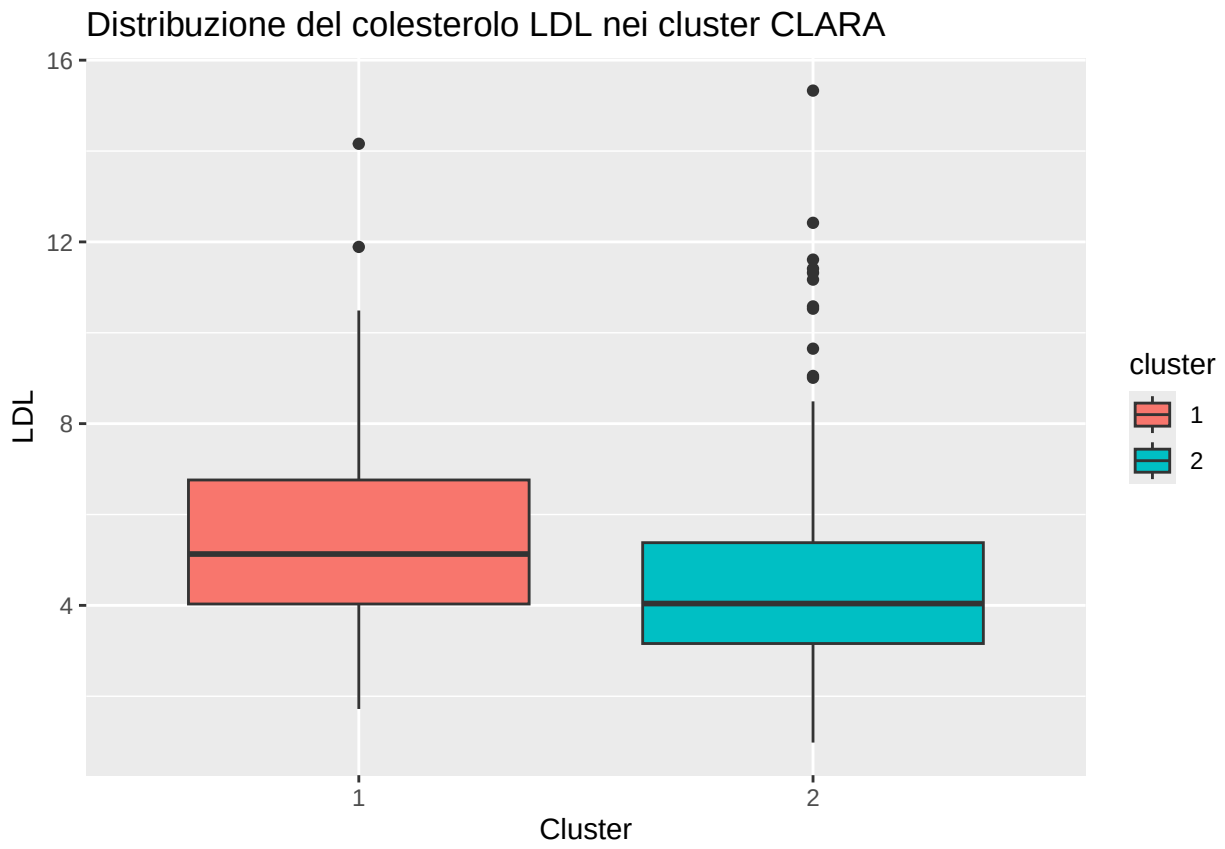
```
                   list(mean = mean, sd = sd, min = min, max = max),
                    na.rm = TRUE))

print(summary_by_cluster)
```

```
## # A tibble: 2 x 41
##   cluster y_mean  y_sd y_min y_max sbp_mean sbp_sd sbp_min sbp_max tobacco_mean
##   <fct>    <dbl> <dbl> <dbl> <dbl>    <dbl>  <dbl>   <dbl>   <dbl>        <dbl>
## 1 1        0.504 0.502     0     1     166.   17.0     144     218         5.31
## 2 2        0.290 0.455     0     1     129.   10.2     101     154         3.04
## # i 31 more variables: tobacco_sd <dbl>, tobacco_min <dbl>, tobacco_max <dbl>,
## #   ldl_mean <dbl>, ldl_sd <dbl>, ldl_min <dbl>, ldl_max <dbl>,
## #   adiposity_mean <dbl>, adiposity_sd <dbl>, adiposity_min <dbl>,
## #   adiposity_max <dbl>, famhist_mean <dbl>, famhist_sd <dbl>,
## #   famhist_min <dbl>, famhist_max <dbl>, typea_mean <dbl>, typea_sd <dbl>,
## #   typea_min <dbl>, typea_max <dbl>, obesity_mean <dbl>, obesity_sd <dbl>,
## #   obesity_min <dbl>, obesity_max <dbl>, alcohol_mean <dbl>, ...
```

```
library(ggplot2)
ggplot(heart_clustered, aes(x = cluster, y = age, fill = cluster)) +
  geom_boxplot() +
  labs(title = "Distribuzione dell'età nei cluster CLARA",
       x = "Cluster",
       y = "Età")
```

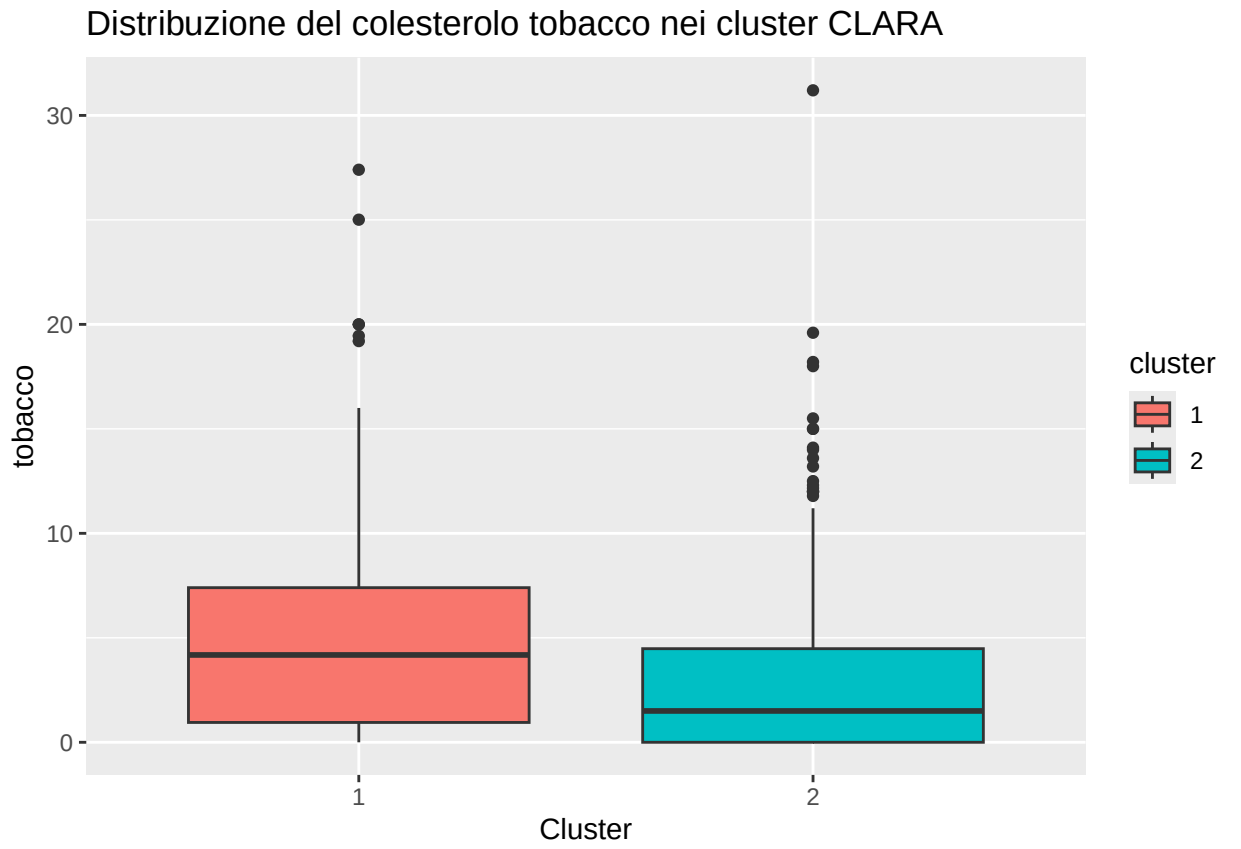

```
ggplot(heart_clustered, aes(x = cluster, y = ldl, fill = cluster)) +
  geom_boxplot() +
```

```
  labs(title = "Distribuzione del colesterolo LDL nei cluster CLARA",
       x = "Cluster",
       y = "LDL")
```




```
ggplot(heart_clustered, aes(x = cluster, y = sbp, fill = cluster)) +
  geom_boxplot() +
  labs(title = "Distribuzione del colesterolo sbp nei cluster CLARA",
       x = "Cluster",
       y = "sbp")
```


```
ggplot(heart_clustered, aes(x = cluster, y = tobacco, fill = cluster)) +
  geom_boxplot() +
  labs(title = "Distribuzione del colesterolo tobacco nei cluster CLARA",
       x = "Cluster",
       y = "tobacco")
```




```
t_age <- t.test(age ~ cluster, data = heart_clustered)
t_ldl <- t.test(ldl ~ cluster, data = heart_clustered)
t_tobacco <- t.test(tobacco ~ cluster, data = heart_clustered)
t_sbp <- t.test(sbp ~ cluster, data = heart_clustered)

# Visualizza i risultati
t_age
```

```
##
##  Welch Two Sample t-test
##
## data:  age by cluster
## t = 12.936, df = 345.3, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  12.25997 16.65659
## sample estimates:
## mean in group 1 mean in group 2
##        53.48760        39.02933
```

```
t_ldl
```

```
##
##  Welch Two Sample t-test
##
## data:  ldl by cluster
## t = 4.8619, df = 196.31, p-value = 2.377e-06
```

```
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  0.6426263 1.5197571
## sample estimates:
## mean in group 1 mean in group 2
##        5.538347        4.457155
```

```
t_tobacco
```

```
##
##  Welch Two Sample t-test
##
## data:  tobacco by cluster
## t = 4.1429, df = 168.6, p-value = 5.415e-05
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  1.186930 3.347781
## sample estimates:
## mean in group 1 mean in group 2
##        5.309174        3.041818
```

```
t_sbp
```

```
##
##  Welch Two Sample t-test
##
## data:  sbp by cluster
## t = 22.719, df = 152, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group 1 and group 2 is not equal to 0
## 95 percent confidence interval:
##  33.97659 40.44889
## sample estimates:
## mean in group 1 mean in group 2
##        165.7934        128.5806
```

I test suggeriscono che il cluster 1 rappresenti pazienti con profili di rischio più elevati: - Età mediamente maggiore. - Pressione sistolica sensibilmente più alta. - Consumo di tabacco superiore. - Colesterolo LDL più alto. Il cluster 2, al contrario, include pazienti più giovani, con pressione sistolica e colesterolo più bassi e un consumo di tabacco ridotto.

```
#Scelta del migliore algoritmo
library(clValid)
library(kohonen)
library(mclust)
clmethods <- c("hierarchical", "kmeans", "diana", "som",
               "model", "sota", "pam", "clara", "agnes")
internal_validation <- clValid(df.scaled, nClust = 2:6, clMethods = clmethods,
                               validation = "internal")

summary(internal_validation)
```

```
##
## Clustering Methods:
##  hierarchical kmeans diana som model sota pam clara agnes
##
## Cluster sizes:
##  2 3 4 5 6
```

```
##
## Validation Measures:
##                                  2        3        4        5        6
##
## hierarchical Connectivity    6.4619  30.3694  33.2984  35.2857  59.2079
##              Dunn            0.0855   0.0617   0.0617   0.0617   0.0558
##              Silhouette      0.5169   0.3747   0.2767   0.2398   0.3054
## kmeans       Connectivity   29.6520  76.3194  63.9397 108.0147 117.5599
##              Dunn            0.0333   0.0219   0.0488   0.0532   0.0513
##              Silhouette      0.4421   0.3763   0.3643   0.3488   0.3189
## diana        Connectivity   25.4687  31.1917  59.3516  78.1921  93.7972
##              Dunn            0.0333   0.0417   0.0425   0.0417   0.0417
##              Silhouette      0.4392   0.3917   0.3746   0.3332   0.2898
## som          Connectivity   27.5750  71.9143  70.4754  88.1198 115.3706
##              Dunn            0.0291   0.0216   0.0298   0.0396   0.0469
##              Silhouette      0.4363   0.3769   0.3624   0.3582   0.3460
## model        Connectivity  147.9163 161.7270 237.1583 340.6881 351.8925
##              Dunn            0.0196   0.0111   0.0218   0.0117   0.0091
##              Silhouette      0.3106   0.2004   0.1099   0.0743   0.0499
## sota         Connectivity   48.3242  62.3909  95.3476 101.2956 106.4548
##              Dunn            0.0210   0.0281   0.0330   0.0336   0.0336
##              Silhouette      0.4125   0.3063   0.3300   0.3178   0.3160
## pam          Connectivity   33.4579  77.1948  94.1171  92.1389 127.5440
##              Dunn            0.0291   0.0219   0.0237   0.0407   0.0373
##              Silhouette      0.4324   0.3766   0.3421   0.3581   0.3390
## clara        Connectivity   28.5683  73.0425 124.4730 142.7754 126.0675
##              Dunn            0.0450   0.0305   0.0219   0.0407   0.0461
##              Silhouette      0.4446   0.3722   0.3379   0.3155   0.3250
## agnes        Connectivity    6.4619  30.3694  33.2984  35.2857  59.2079
##              Dunn            0.0855   0.0617   0.0617   0.0617   0.0558
##              Silhouette      0.5169   0.3747   0.2767   0.2398   0.3054
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.4619 hierarchical 2
## Dunn         0.0855 hierarchical 2
## Silhouette   0.5169 hierarchical 2
```

```
print(optimalScores(internal_validation))
```

```
##                   Score       Method Clusters
## Connectivity 6.46190476 hierarchical        2
## Dunn         0.08552368 hierarchical        2
## Silhouette   0.51686021 hierarchical        2
```

```
## Stability measures
clmethods <- c("hierarchical", "kmeans", "diana", "som",
               "model", "sota", "pam", "clara", "agnes")
stab_validation <- clValid(df.scaled, nClust = 2:6, clMethods = clmethods,
                validation = "stability")

summary(stab_validation)
```

```
##
```

```
## Clustering Methods:
##  hierarchical kmeans diana som model sota pam clara agnes
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                          2       3       4       5       6
##
## hierarchical APN   0.0645  0.1887  0.3278  0.3343  0.3444
##              AD   29.4985 24.9556 24.7043 24.5105 22.4545
##              ADM   4.2767  7.7791  9.9698 10.0959 12.6142
##              FOM  10.3119  9.7915  9.6632  9.6620  9.6202
## kmeans       APN   0.1169  0.1088  0.2516  0.3452  0.2840
##              AD   25.0708 22.0762 20.9224 20.0431 18.4772
##              ADM   5.2653  7.1843  8.0927 10.1948  7.5908
##              FOM   9.7200  9.7028  9.6849  9.5509  9.6085
## diana        APN   0.1193  0.1212  0.2102  0.2021  0.2536
##              AD   25.0947 23.9089 20.7246 19.7758 18.8100
##              ADM   5.3508  5.3302  6.8405  7.2994  7.2229
##              FOM   9.7222  9.7288  9.6077  9.5955  9.5726
## som          APN   0.1232  0.2442  0.2889  0.3426  0.3094
##              AD   25.1166 22.0660 20.9308 19.7323 18.3486
##              ADM   5.3753  7.7176  8.6677  9.3409  7.7789
##              FOM   9.7425  9.6180  9.6527  9.5775  9.5594
## model        APN   0.2058  0.2374  0.3532  0.4199  0.4034
##              AD   28.0546 25.2082 25.3761 25.0047 24.2009
##              ADM   8.4467  6.5627  9.6748  9.4818 10.2229
##              FOM   9.8859  9.6279  9.5261  9.4917  9.4927
## sota         APN   0.1211  0.1503  0.1813  0.1877  0.2666
##              AD   25.0378 23.3727 20.5021 19.8393 19.5176
##              ADM   5.1565  5.1073  6.7652  6.5155  6.6246
##              FOM   9.7020  9.6775  9.6143  9.6063  9.5817
## pam          APN   0.1228  0.2189  0.2615  0.3333  0.4042
##              AD   25.0461 21.8072 20.6851 19.6709 19.0214
##              ADM   5.1228  7.0748  8.5616  8.7933  9.6288
##              FOM   9.7295  9.6226  9.6198  9.6030  9.5283
## clara        APN   0.1330  0.2127  0.2817  0.4958  0.4072
##              AD   25.2101 21.8261 20.9301 21.2716 19.3250
##              ADM   5.6108  6.7410  8.4966 12.6851  9.7565
##              FOM   9.7361  9.6137  9.6462  9.5647  9.5726
## agnes        APN   0.0645  0.1887  0.3278  0.3343  0.3444
##              AD   29.4985 24.9556 24.7043 24.5105 22.4545
##              ADM   4.2767  7.7791  9.9698 10.0959 12.6142
##              FOM  10.3119  9.7915  9.6632  9.6620  9.6202
##
## Optimal Scores:
##
##     Score   Method       Clusters
## APN  0.0645 hierarchical 2
## AD  18.3486 som          6
## ADM  4.2767 hierarchical 2
## FOM  9.4917 model        5
```

```r
print(optimalScores(stab_validation))
```

```
##           Score       Method Clusters
## APN  0.06452207 hierarchical        2
## AD  18.34863050          som        6
## ADM  4.27673997 hierarchical        2
## FOM  9.49167779        model        5
```

L'analisi dei metodi di clustering è stata condotta mediante una serie di metriche di validazione interna per valutare la qualità delle diverse tecniche di partizionamento. Le misure di Silhouette, Dunn Index e Connectivity sono state calcolate per confrontare vari algoritmi, tra cui Hierarchical Clustering, K-Means, DIANA, SOM, PAM, CLARA e AGNES.

I risultati indicano che il metodo di clustering gerarchico (hierarchical clustering) con 2 cluster ha ottenuto i punteggi più elevati per tutte le metriche di validazione:

Silhouette Score = 0.5169, il valore più alto tra tutte le tecniche testate, indicando una buona separazione tra i cluster.

Dunn Index = 0.0855, suggerendo una distanza inter-cluster relativamente elevata rispetto alla dispersione intra-cluster.

Connectivity = 6.4619, il valore più basso tra gli approcci testati, indicando che i punti all'interno di ciascun cluster sono ben connessi tra loro.

Queste metriche suggeriscono che il clustering gerarchico con 2 gruppi è la soluzione ottimale per segmentare il dataset in base alla sua struttura intrinseca. Le tecniche alternative, come il K-Means o il clustering basato su densità, mostrano punteggi di silhouette più bassi e valori di connectivity più elevati, suggerendo una qualità di partizione inferiore.

Questi risultati supportano l'ipotesi che il dataset contenga due gruppi distinti con caratteristiche ben separate, rendendo il clustering gerarchico l'approccio più adatto per l'analisi. Ulteriori analisi potrebbero includere metriche di validazione esterna o l'utilizzo di metodi di clustering più avanzati, come i Gaussian Mixture Models (GMM), per confermare la robustezza di questa segmentazione.

L'analisi di stabilità clustering è stata valutata attraverso diverse metriche di validazione, tra cui APN (Average Proportion of Non-overlap), AD (Average Distance), ADM (Average Density Mean) e FOM (Figure of Merit), applicate su vari metodi di clustering come Hierarchical Clustering, K-Means, DIANA, SOM, SOTA, PAM, CLARA e AGNES.

I risultati evidenziano che:

Il clustering gerarchico (Hierarchical Clustering) con 2 cluster ha ottenuto il valore più basso di APN = 0.0645, suggerendo una maggiore stabilità dei cluster e una minore sovrapposizione tra le osservazioni assegnate.

Il metodo SOM con 6 cluster ha il valore più alto di AD = 18.3486, il che indica cluster ben separati.

Il clustering gerarchico con 2 cluster ha ottenuto il valore più basso per ADM = 4.2767, indicando una buona coesione interna dei cluster.

Il modello model con 6 cluster ha il valore più basso per FOM = 9.4917, suggerendo una minimizzazione dell'errore di classificazione rispetto ai dati originali.

Questi risultati indicano che la soluzione ottimale dipende dall'obiettivo dell'analisi. Se si cerca stabilità e coesione interna, il clustering gerarchico con 2 cluster sembra essere la scelta più appropriata.

In sintesi, il clustering gerarchico è ideale per cluster stabili e ben separati.

```r
library(caret)
# Separare la variabile target (y) e le variabili indipendenti
y_true <- heart[, 1]   # Variabile binaria target (1,2)
```

```r
# Clustering Gerarchico
hc_res <- hclust(dist_matrix, method = "ward.D2")

# Taglio del dendrogramma per ottenere 2 cluster
clusters <- cutree(hc_res, k = 2)

# Aggiunge al dataset heart i cluster trovati
heart_clustered <- heart[,vars]
heart_clustered$cluster <- factor(clusters)  # Assegna i cluster

library(dplyr)
summary_by_cluster <- heart_clustered %>%
  group_by(cluster) %>%
  summarise(across(where(is.numeric),
                   list(mean = mean, sd = sd, min = min, max = max),
                    na.rm = TRUE))

print(summary_by_cluster)
```

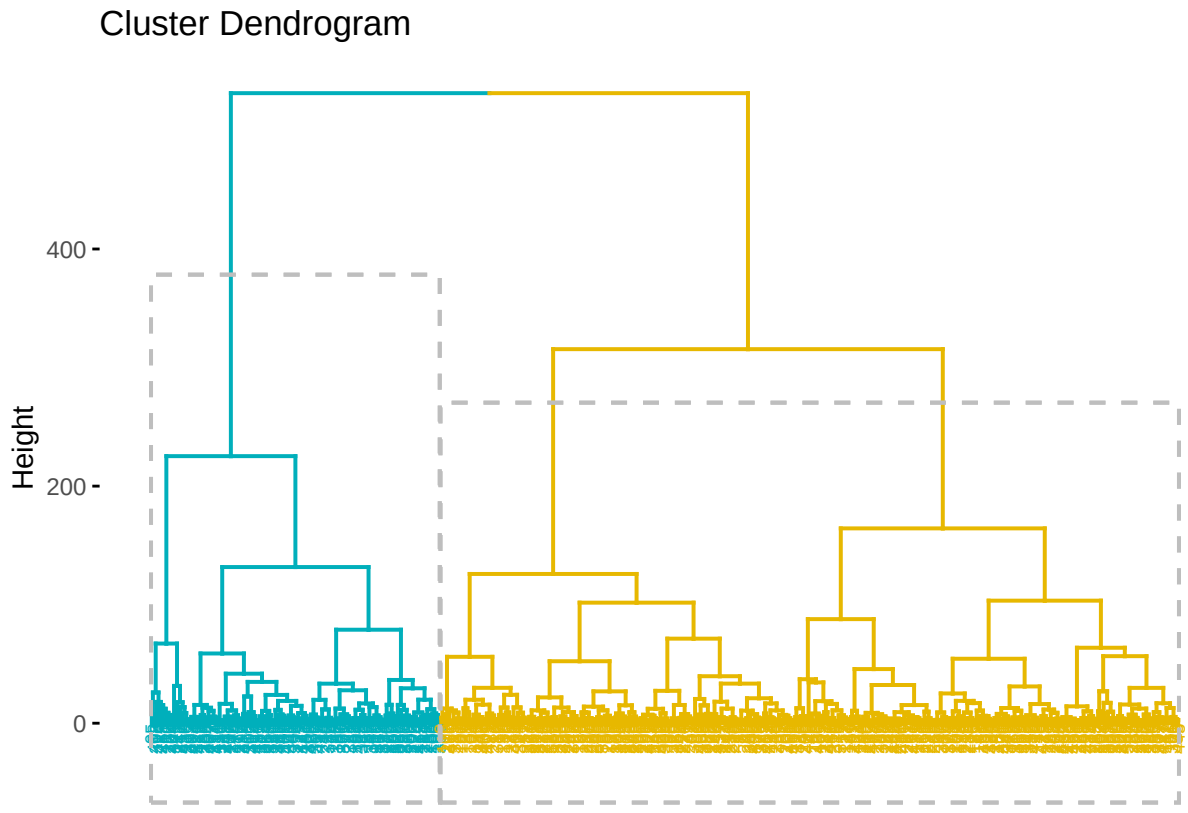
```
## # A tibble: 2 x 17
##   cluster obesity_mean obesity_sd obesity_min obesity_max ldl_mean ldl_sd
##   <fct>          <dbl>      <dbl>       <dbl>       <dbl>    <dbl>  <dbl>
## 1 1               27.8       4.16        20.4        41.8     5.57   2.14
## 2 2               25.4       4.04        14.7        46.6     4.42   1.95
## # i 10 more variables: ldl_min <dbl>, ldl_max <dbl>, age_mean <dbl>,
## #   age_sd <dbl>, age_min <dbl>, age_max <dbl>, sbp_mean <dbl>, sbp_sd <dbl>,
## #   sbp_min <dbl>, sbp_max <dbl>
```

```r
# Valutazione della qualità del clustering (il cluster 1 è quello associato
# al valore medio più alto delle variabile associate a rischio di malattia cardiaca)

confusion <- confusionMatrix(as.factor(clusters==1), as.factor(y_true==1))

# Stampiamo le metriche di valutazione
print(confusion)
```

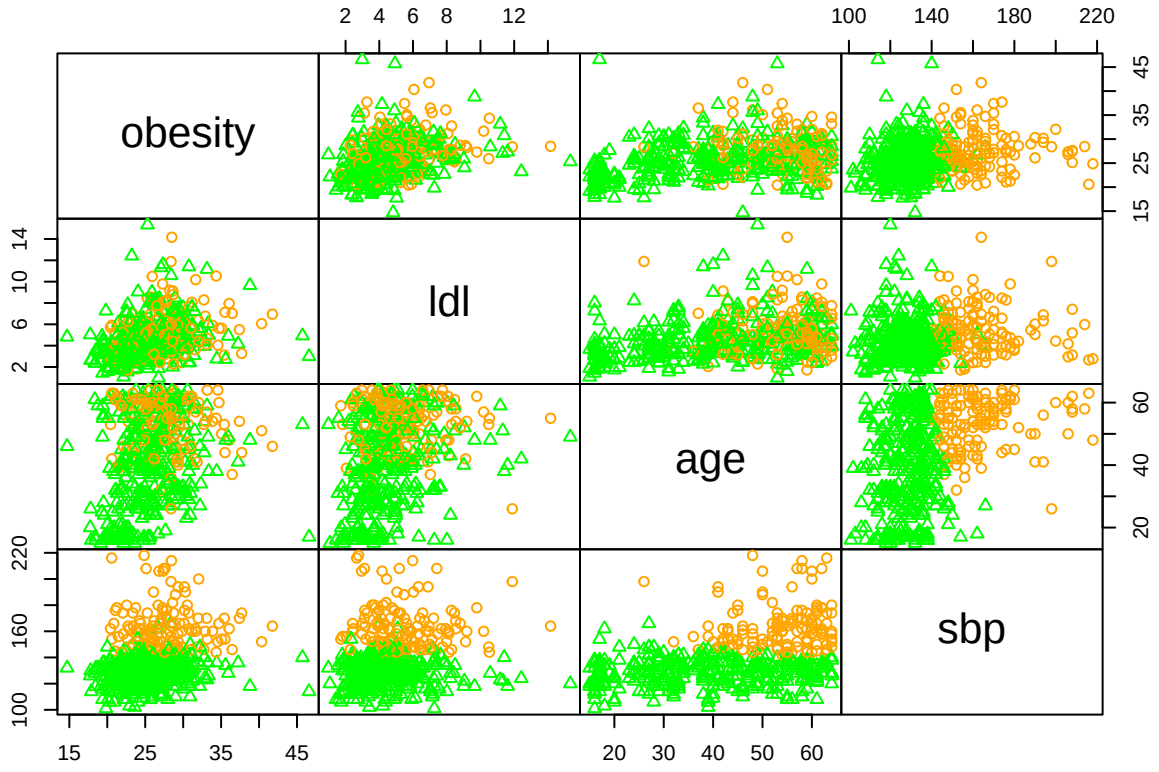
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   238   94
##      TRUE     64   66
##
##                Accuracy : 0.658
##                  95% CI : (0.6128, 0.7012)
##     No Information Rate : 0.6537
##     P-Value [Acc > NIR] : 0.44364
##
##                   Kappa : 0.2098
##
##  Mcnemar's Test P-Value : 0.02105
##
##             Sensitivity : 0.7881
##             Specificity : 0.4125
```

```
##          Pos Pred Value : 0.7169
##          Neg Pred Value : 0.5077
##              Prevalence : 0.6537
##          Detection Rate : 0.5152
##    Detection Prevalence : 0.7186
##       Balanced Accuracy : 0.6003
##
##        'Positive' Class : FALSE
##
```

```r
fviz_dend(hc_res, k = 2, # Cut in 2 groups
          cex = 0.5, # label size
          k_colors = c("#00AFBB", "#E7B800"),
          color_labels_by_k = TRUE, # color labels by groups
          rect = TRUE ) # Add rectangle around groups
```




```r
# Matrice di scatterplots delle variabili obesity, ldl, sbp, age
pairs(heart_vars, gap=0, pch=clusters, col=c("orange", "green")[clusters])
```



Per scegliere le variabili migliori per applicazione del modello GMM facciamo alcune considerazioni sulle variabili che possono avere: - Buona separazione tra cluster negli scatter plot - Distribuzione potenzialmente multimodale - Bassa collinearità (evitare variabili troppo correlate)

Scelte migliori per GMM 1. Età (age) Ha una distribuzione che mostra più gruppi distinti È correlata a sbp e obesity, ma non eccessivamente

2. Obesità (obesity) Alta correlazione con adiposity, quindi scegliamo solo una delle due Mostra cluster visibili
3. LDL (ldl) Indicatore clinico importante, mostra qualche separazione tra gruppi
4. Pressione Sistolica (sbp) Correlata a età, ma mantiene una distribuzione interessante

Variabili da escludere: Tipo A (typea) → Non mostra una separazione chiara Alcol (alcohol) → Molta dispersione e poche differenze tra cluster Tabacco (tobacco) → Distribuzione poco informativa

Le migliori variabili per un modello di Mixture Gaussiane sono: 1. Età (age) 2. Obesità (obesity) (o adiposità, ma non entrambe) 3. LDL (ldl) 4. Pressione Sistolica (sbp)

```r
# Caricamento delle librerie
library(mclust)
library(GGally)

# Applicazione del Gaussian Mixture Model
set.seed(123)
gmm_model <- Mclust(heart[, c("age", "obesity", "ldl", "sbp")], G = 2)

# Risultati del modello
summary(gmm_model)
```

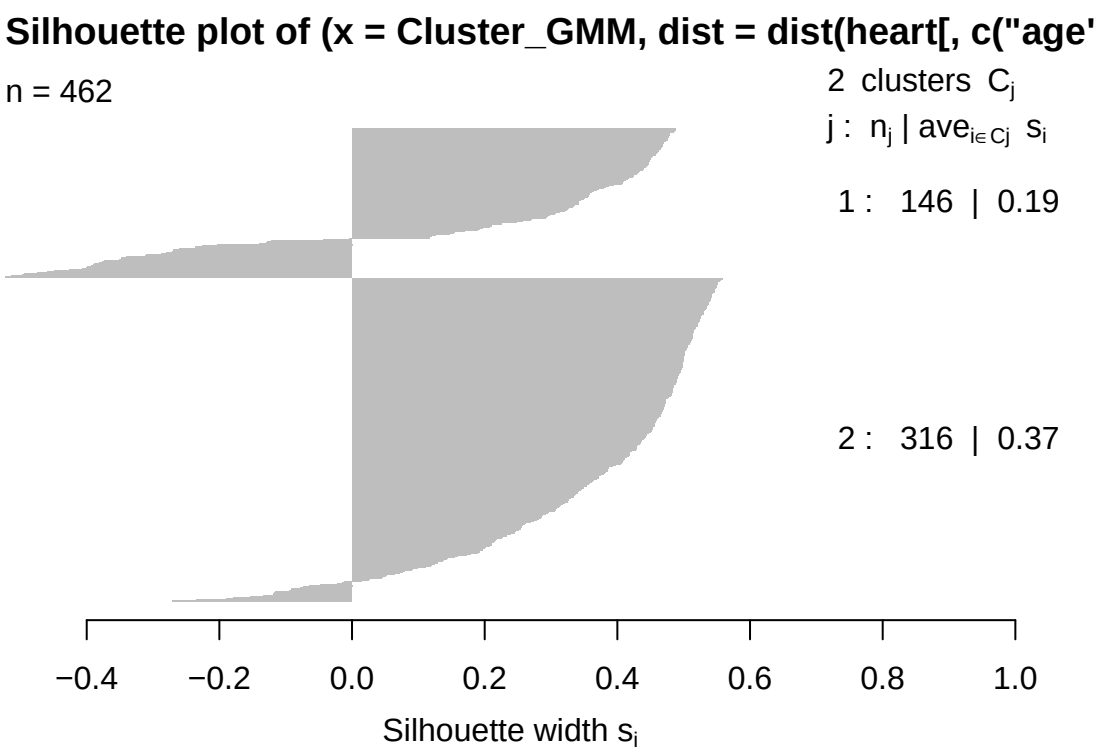
```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust VVV (ellipsoidal, varying volume, shape, and orientation) model with 2
## components:
##
##  log-likelihood   n df      BIC       ICL
##       -6004.134 462 29 -12186.2 -12297.67
##
## Clustering table:
##   1   2
## 146 316
```

```r
# Predizioni dei cluster
Cluster_GMM <- gmm_model$classification

C1 <- heart_vars[Cluster_GMM == "1", ]
C2 <- heart_vars[Cluster_GMM == "2", ]

# Creazione della tabella con medie, deviazioni standard e dimensione del cluster
df_cluster <- data.frame(
  "Media Cluster 1" = apply(C1, 2, mean),  # Medie Cluster 1
  "Deviazione Std Cluster 1" = apply(C1, 2, sd),  # Deviazione Standard Cluster 1
  "N Cluster 1" = nrow(C1),  # Numero di osservazioni Cluster 1
  "Media Cluster 2" = apply(C2, 2, mean),  # Medie Cluster 2
  "Deviazione Std Cluster 2" = apply(C2, 2, sd),  # Deviazione Standard Cluster 2
  "N Cluster 2" = nrow(C2)  # Numero di osservazioni Cluster 2
)

library(knitr)
kable(df_cluster, caption = "Profili dei Cluster")
```

Table 3: Profili dei Cluster

	Media.Cluster.1	Deviazione.Std.Cluster.1	N.Cluster.1	Media.Cluster.2	Deviazione.Std.Cluster.2	N.Cluster.2
obesity	28.39103	4.964302	146	24.959778	3.302699	316
ldl	6.19863	2.525768	146	4.066551	1.377944	316
age	52.85616	8.320303	146	38.177215	14.565896	316
sbp	157.23973	22.946205	146	129.588608	11.404054	316

```r
# Il Cluster 1 rappresenta individui a più alto rischio cardiovascolare → età più avanzata, maggiore ob
# Il Cluster 2 rappresenta individui più giovani e metabolicamente più sani, con valori più bassi di pr

# Calcolo del silhouette_score
silhouette_scores <- silhouette(Cluster_GMM,
                                dist(heart[, c("age", "obesity", "ldl", "sbp")]))
plot(silhouette_scores)
```



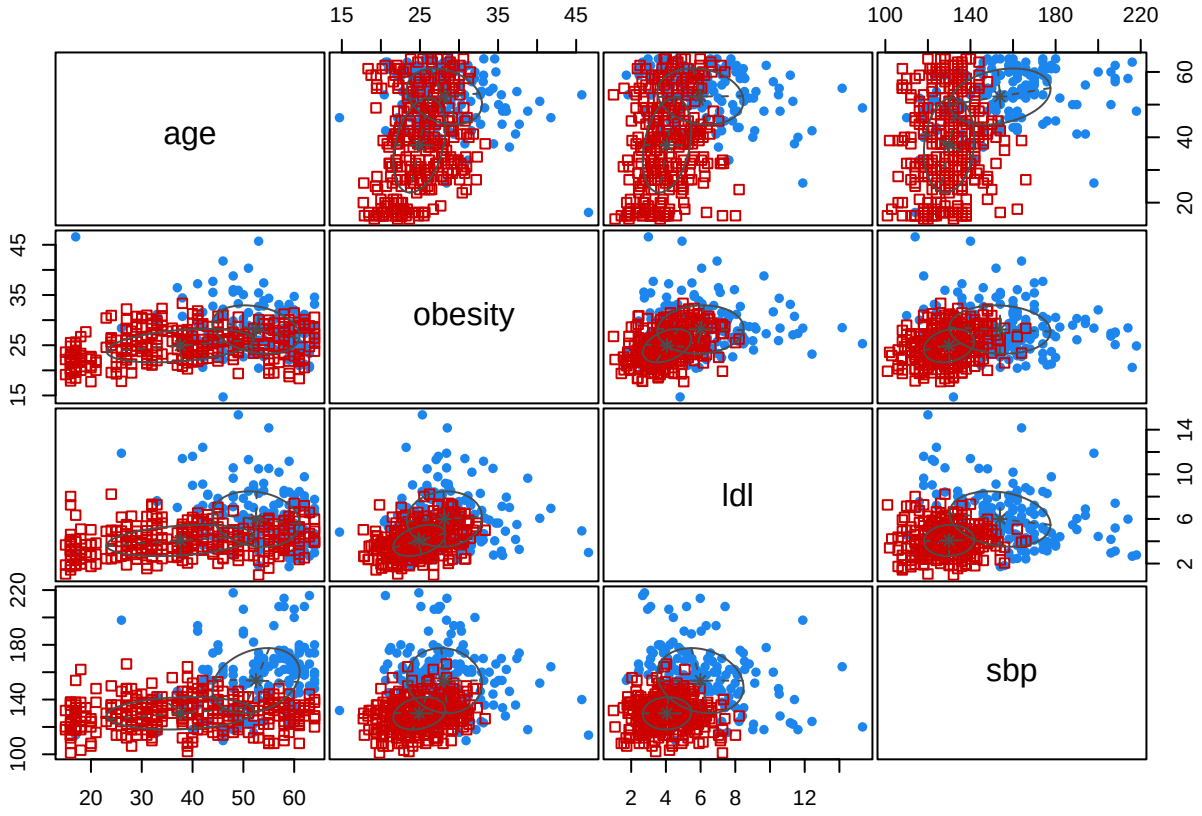
Average silhouette width : 0.31

```
# Valutazione della qualità del clustering
confusion <- confusionMatrix(as.factor(Cluster_GMM ==1),
                             as.factor(y_true==1))

# Stampiamo le metriche di valutazione
print(confusion)
```

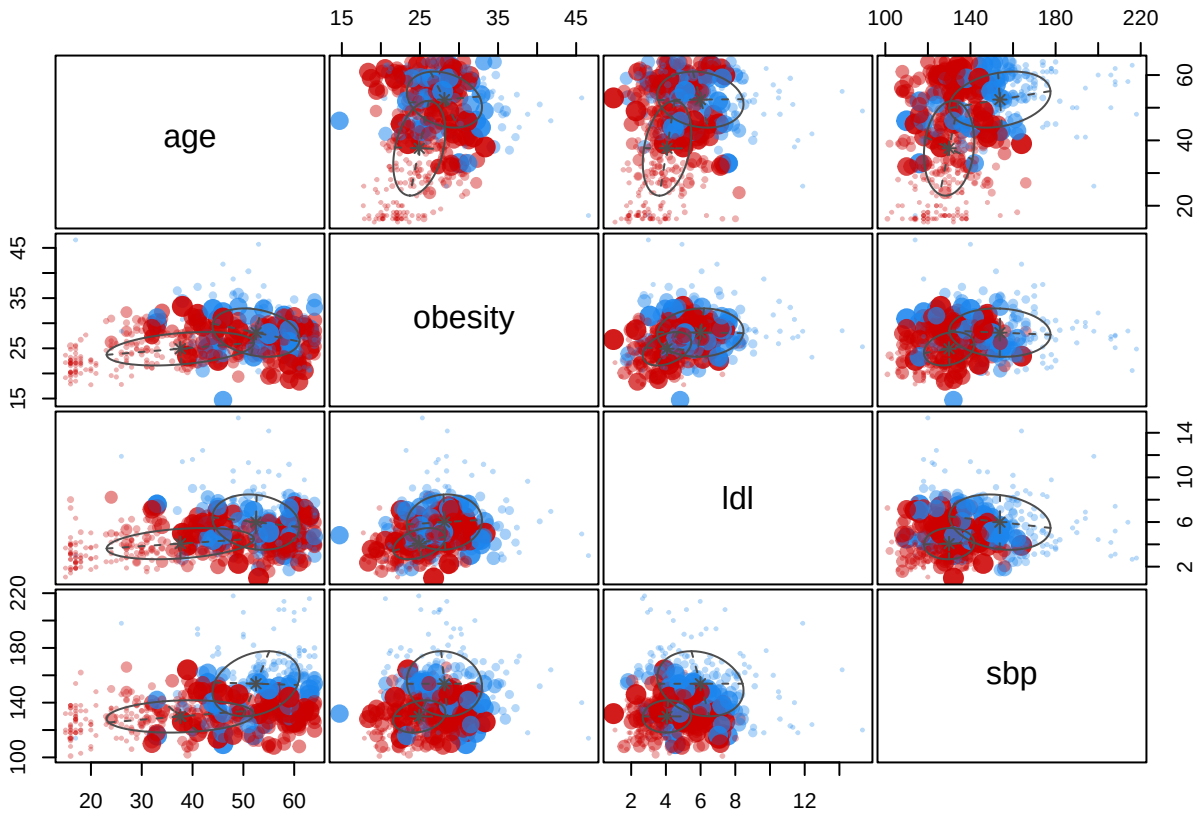
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   233   83
##      TRUE     69   77
##
##                Accuracy : 0.671
##                  95% CI : (0.6261, 0.7137)
##     No Information Rate : 0.6537
##     P-Value [Acc > NIR] : 0.2324
##
##                   Kappa : 0.2581
##
##  Mcnemar's Test P-Value : 0.2917
##
##             Sensitivity : 0.7715
##             Specificity : 0.4813
##          Pos Pred Value : 0.7373
```

```
##          Neg Pred Value : 0.5274
##              Prevalence : 0.6537
##          Detection Rate : 0.5043
##    Detection Prevalence : 0.6840
##       Balanced Accuracy : 0.6264
##
##        'Positive' Class : FALSE
##
```

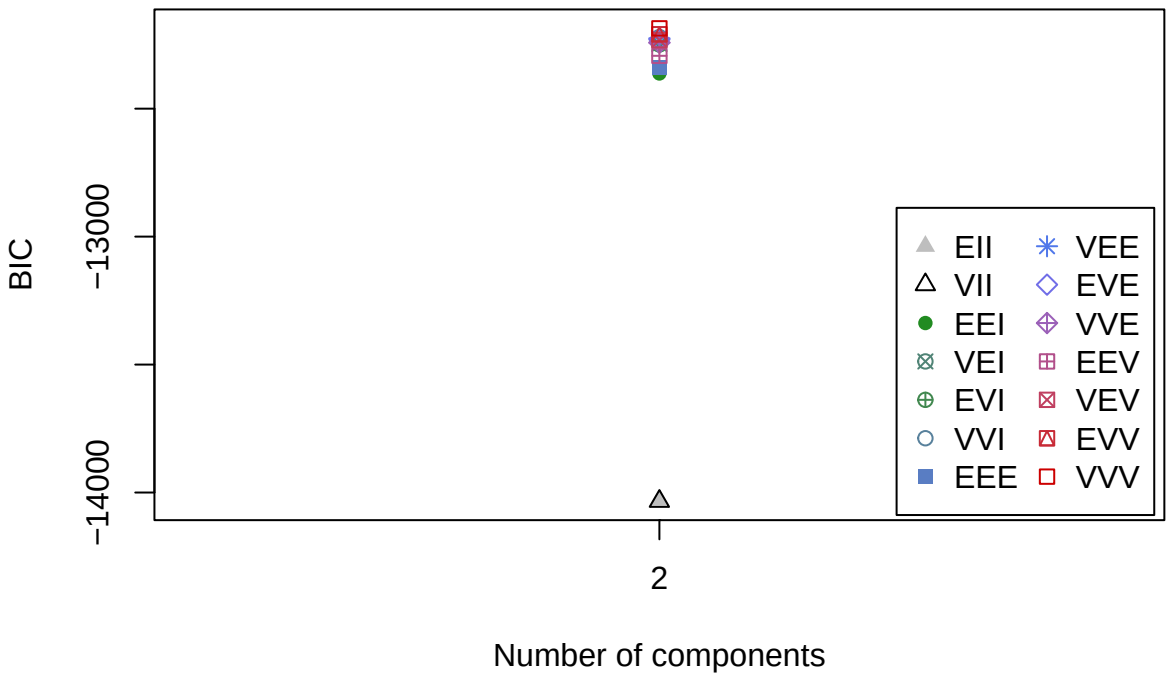
```r
# Visualizzazione della distribuzione dei cluster
plot(gmm_model, what = "classification")  # Mostra la classificazione
```




```r
# Visualizzazione della probabilità di appartenenza ai cluster
plot(gmm_model, what = "uncertainty")
```




```r
# Visualizzazione dei cluster nei primi due componenti principali (PCA)
plot(gmm_model, what = "BIC")  # Confronto dei modelli con criterio BIC
```




```
print(gmm_model$BIC)
```

```
## Bayesian Information Criterion (BIC):
##         EII       VII       EEI       VEI       EVI       VVI       EEE
## 2 -14029.02 -14034.01 -12362.46 -12247.07 -12249.31 -12219.94 -12339.16
##         VEE       EVE       VVE       EEV       VEV      EVV      VVV
## 2 -12224.99 -12229.37 -12241.99 -12293.44 -12231.84 -12209.2 -12186.2
##
## Top 3 models based on the BIC criterion:
##     VVV,2     EVV,2     VVI,2
## -12186.20 -12209.20 -12219.94
```

Il modello di Mixture Gaussiane (GMM) con due componenti, stimato sulle variabili età, obesità, ldl e sbp, descrive in maniera efficace due gruppi ben distinti per profilo clinico. Il primo cluster, più ridotto nelle dimensioni, riunisce individui mediamente più anziani, con livelli più elevati di colesterolo LDL, pressione sistolica e obesità, suggerendo un quadro clinico potenzialmente più "a rischio". Il secondo, invece, è costituito da persone più giovani, meno obese, con pressione e colesterolo sensibilmente inferiori.

Nonostante questa suddivisione rifletta una chiara diversità nei fattori di rischio, il confronto con la variabile binaria "malattia sì/no" rivela un'accuratezza soltanto moderata (circa il 67%). Il cluster "ad alto rischio" ingloba sì numerosi pazienti realmente malati, ma anche un discreto numero di soggetti che, pur mostrando profili simili, non risultano affetti da patologie coronariche. Ciò si traduce in una specificità piuttosto contenuta, indice del fatto che avere fattori di rischio elevati non coincide necessariamente con l'aver sviluppato la malattia.

Questo risultato, comunque, non inficia il valore descrittivo del GMM: l'obiettivo di un modello a mixture non è la classificazione supervisionata, bensì individuare componenti gaussiane latenti che riassumano la variabilità del campione. In tal senso, il clustering differenzia in modo netto chi presenta combinazioni di

età, pressione e colesterolo più sfavorevoli, dai pazienti con parametri mediamente bassi. Rimane comunque evidente come i fattori considerati (età, sbp, ldl, obesità) traccino due sottopopolazioni distinte, confermando la loro rilevanza per definire un profilo di rischio, ancorché non perfettamente sovrapponibile alla presenza effettiva di patologie cardiache.

Il modello "VVV" con 2 cluster ha il BIC più basso, il che significa che è il migliore in termini di compromesso tra accuratezza e complessità. Questo modello permette variazioni di volume, forma e orientamento tra i cluster, suggerendo una struttura di cluster complessa.

```
library(catdata)
library(MASS)
library(caret)
library(pROC)
library(biotools)
library(mvnormtest)
library(car)

data(heart)

heart <- data.frame(heart)

# Conversione della variabile target in fattore
heart$y <- factor(heart$y, levels = c(0,1), labels = c("No_CHD", "CHD"))

# Conversione della variabile famhist in fattore
heart$famhist <- factor(heart$famhist, levels = c(0,1), labels = c("Absent", "Present"))

str(heart)
```

```
## 'data.frame':    462 obs. of  10 variables:
##  $ y        : Factor w/ 2 levels "No_CHD","CHD": 2 2 1 2 2 1 1 2 1 2 ...
##  $ sbp      : num  160 144 118 170 134 132 142 114 114 132 ...
##  $ tobacco  : num  12 0.01 0.08 7.5 13.6 6.2 4.05 4.08 0 0 ...
##  $ ldl      : num  5.73 4.41 3.48 6.41 3.5 6.47 3.38 4.59 3.83 5.8 ...
##  $ adiposity: num  23.1 28.6 32.3 38 27.8 ...
##  $ famhist  : Factor w/ 2 levels "Absent","Present": 2 1 2 2 2 2 1 2 2 2 ...
##  $ typea    : num  49 55 52 51 60 62 59 62 49 69 ...
##  $ obesity  : num  25.3 28.9 29.1 32 26 ...
##  $ alcohol  : num  97.2 2.06 3.81 24.26 57.34 ...
##  $ age      : num  52 63 46 58 49 45 38 58 29 53 ...
```

```
# Divisione dataset in training (70%) e test set (30%)
set.seed(123)
train_index <- createDataPartition(heart$y, p = 0.7, list = FALSE)
heart_train <- heart[train_index, ]
heart_test <- heart[-train_index, ]

# LDA assume distribuzione Normale delle Classi e Varianza-Covarianza Uguale tra le Classi tra classi.
heart_CHD <- subset(heart, y=="CHD")[,-1] # Escludiamo la variabile risposta
heart_noCHD <- subset(heart, y=="No_CHD")[,-1]

#Test di Shapiro-Wilk per normalità univariata
# Normalità univariata variabili numeriche:
apply(heart_CHD[, -5], 2, shapiro.test)  # esclusa famhist (categorica)
```

```
## $sbp
```

```
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.93846, p-value = 2.074e-06
##
##
## $tobacco
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.83919, p-value = 5.652e-12
##
##
## $ldl
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.92866, p-value = 3.902e-07
##
##
## $adiposity
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.98559, p-value = 0.09639
##
##
## $typea
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.99247, p-value = 0.5697
##
##
## $obesity
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.96239, p-value = 0.0002468
##
##
## $alcohol
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.73589, p-value = 1.212e-15
##
```

```
##
## $age
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.92273, p-value = 1.514e-07
```

```
apply(heart_noCHD[, -5], 2, shapiro.test)
```

```
## $sbp
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.92895, p-value = 8.077e-11
##
##
## $tobacco
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.74791, p-value < 2.2e-16
##
##
## $ldl
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.91641, p-value = 6.191e-12
##
##
## $adiposity
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.98136, p-value = 0.0005734
##
##
## $typea
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.98491, p-value = 0.002932
##
##
## $obesity
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
```

```
## W = 0.95766, p-value = 1.119e-07
##
##
## $alcohol
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.70296, p-value < 2.2e-16
##
##
## $age
##
##  Shapiro-Wilk normality test
##
## data:  newX[, i]
## W = 0.94511, p-value = 3.56e-09
```

```
#Test di normalità multivariata di Royston
# Test multivariato (richiede variabili numeriche):
numeric_vars <- c("sbp", "tobacco", "ldl", "adiposity", "typea", "obesity", "alcohol", "age")

# Gruppo CHD
mshapiro.test(t(heart_CHD[, numeric_vars]))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.91752, p-value = 6.816e-08
```

```
# Gruppo No_CHD
mshapiro.test(t(heart_noCHD[, numeric_vars]))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.83329, p-value < 2.2e-16
```

Dai risultati si evince chiaramente che la maggior parte delle variabili presentano una forte violazione dell'ipotesi di normalità univariata, poiché molti valori del p-value risultano minori di 0.05, talvolta addirittura minori di 2.2e-16, evidenziando così significative deviazioni dalla distribuzione normale. Solo "Adiposity" e "TypeA" (nel primo test) mostrano normalità (p-value > 0.05). Tutte le altre variabili non seguono una distribuzione normale. Tuttavia, l'Analisi Discriminante Lineare (LDA) e Quadratica (QDA) richiedono la verifica dell'ipotesi più restrittiva di normalità delle variabili all'interno delle classi.

```
#Test di normalità multivariata di Royston
# Test multivariato (richiede variabili numeriche):
numeric_vars <- c("sbp", "tobacco", "ldl", "adiposity", "typea", "obesity", "alcohol", "age")

# Gruppo CHD
mshapiro.test(t(heart_CHD[, numeric_vars]))
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  Z
## W = 0.91752, p-value = 6.816e-08
```

```
# Gruppo No_CHD
mshapiro.test(t(heart_noCHD[, numeric_vars]))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  Z
## W = 0.83329, p-value < 2.2e-16
```

Questi risultati confermano fortemente la violazione dell'ipotesi di normalità multivariata.

```
#Controllo Omogeneità delle Matrici di Covarianza (Box's M Test):
library(biotools)

boxM(heart[, numeric_vars], heart$y)
```

```
##
##  Box's M-test for Homogeneity of Covariance Matrices
##
## data:  heart[, numeric_vars]
## Chi-Sq (approx.) = 150.49, df = 36, p-value = 6.03e-16
```

p-value < 0.05 indica una significativa differenza delle covarianze, suggerendo l'utilizzo di QDA.

```
#Analisi Discriminante Lineare (LDA)

#Applicazione modello LDA
lda_model <- lda(y ~ ., data = heart_train)
lda_model
```

```
## Call:
## lda(y ~ ., data = heart_train)
##
## Prior probabilities of groups:
##   No_CHD      CHD
## 0.654321 0.345679
##
## Group means:
##             sbp  tobacco      ldl adiposity famhistPresent    typea  obesity
## No_CHD 135.7547 2.726651 4.409151  24.04632      0.3160377 51.83019 25.86566
## CHD    143.3304 5.780804 5.615446  28.85777      0.5892857 54.95536 27.04268
##         alcohol     age
## No_CHD 15.08656 38.8066
## CHD    17.01929 50.5000
##
## Coefficients of linear discriminants:
##                         LD1
## sbp             0.001246965
## tobacco         0.090606282
## ldl             0.151263369
## adiposity       0.041034433
## famhistPresent  0.829337083
## typea           0.032217167
## obesity        -0.074595974
```

```
## alcohol        -0.002196700
## age             0.028072029
```

```
#Valutazione performance LDA (train set)
lda_pred_train <- predict(lda_model, heart_train)
confusionMatrix(lda_pred_train$class, heart_train$y)
```

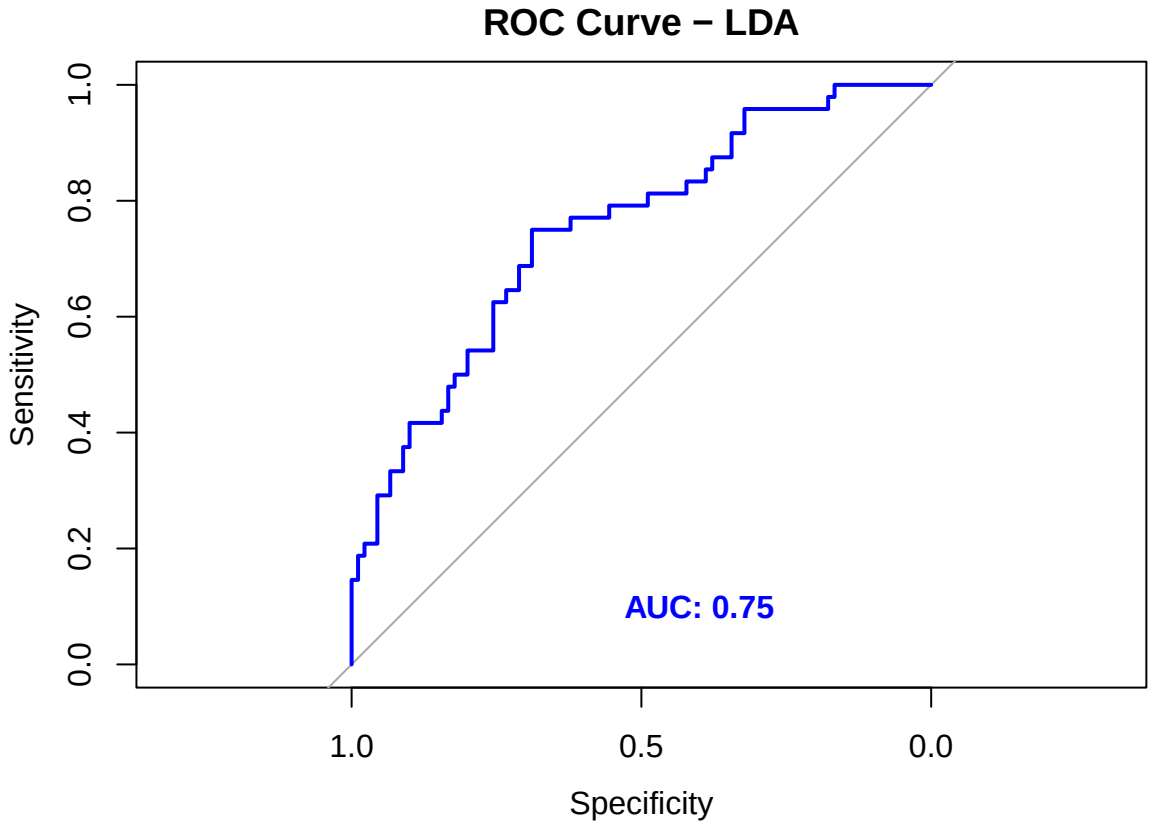
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No_CHD CHD
##     No_CHD    183  53
##     CHD        29  59
##
##                Accuracy : 0.7469
##                  95% CI : (0.6959, 0.7933)
##     No Information Rate : 0.6543
##     P-Value [Acc > NIR] : 0.0002106
##
##                   Kappa : 0.4108
##
##  Mcnemar's Test P-Value : 0.0110876
##
##             Sensitivity : 0.8632
##             Specificity : 0.5268
##          Pos Pred Value : 0.7754
##          Neg Pred Value : 0.6705
##              Prevalence : 0.6543
##          Detection Rate : 0.5648
##    Detection Prevalence : 0.7284
##       Balanced Accuracy : 0.6950
##
##        'Positive' Class : No_CHD
##
```

```
#Valutazione performance LDA (test set)
lda_pred_test <- predict(lda_model, heart_test)
confusionMatrix(lda_pred_test$class, heart_test$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No_CHD CHD
##     No_CHD     73  24
##     CHD        17  24
##
##                Accuracy : 0.7029
##                  95% CI : (0.6192, 0.7776)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.1219
##
##                   Kappa : 0.3221
##
##  Mcnemar's Test P-Value : 0.3487
##
```

```
##             Sensitivity : 0.8111
##             Specificity : 0.5000
##          Pos Pred Value : 0.7526
##          Neg Pred Value : 0.5854
##              Prevalence : 0.6522
##          Detection Rate : 0.5290
##    Detection Prevalence : 0.7029
##       Balanced Accuracy : 0.6556
##
##        'Positive' Class : No_CHD
##
```

```r
#ROC curve e AUC per LDA
roc_lda <- roc(heart_test$y, lda_pred_test$posterior[,2])
auc_value= auc(roc_lda)
plot(roc_lda, main="ROC Curve - LDA", col="blue")
text(0.4, 0.1, paste("AUC:", round(auc_value, 3)), col = "blue", font = 2)
```



Nonostante le violazioni delle assunzioni, ho applicato LDA. La variabile con maggior peso discriminante è la storia familiare ("famhist"), seguita da ldl e tobacco. Queste variabili sembrano avere un impatto significativo sulla presenza di CHD. La variabile obesity appare con segno negativo, suggerendo una relazione inversa (ERRORE DOVUTO AL NON RISPETTO DELLE ASSUNZIONI). Il modello mostra un buon potere predittivo ma una sensibilità (81.11%) più elevata della specificità (50%), indicando che il modello è efficace nell'identificare chi non ha la malattia (No_CHD) ma meno efficace nel riconoscere correttamente chi invece è affetto dalla patologia (CHD).

L'AUC indica una capacità discriminante accettabile/moderata, non elevata (una AUC ottimale è generalmente >0.8).

La curva ROC mostra che il modello è significativamente superiore rispetto alla classificazione casuale (AUC = 0.50), ma la performance non è eccellente.

Il modello LDA applicato risulta non ideale, confermato anche da un'AUC non elevatissima.

```
#Analisi Discriminante Quadratica (QDA)
#QDA non assume uguaglianza delle matrici di covarianza tra classi ed è più flessibile.

#Applicazione modello QDA
qda_model <- qda(y ~ ., data = heart_train)
qda_model
```

```
## Call:
## qda(y ~ ., data = heart_train)
##
## Prior probabilities of groups:
##   No_CHD      CHD
## 0.654321 0.345679
##
## Group means:
##             sbp  tobacco      ldl adiposity famhistPresent    typea  obesity
## No_CHD 135.7547 2.726651 4.409151  24.04632      0.3160377 51.83019 25.86566
## CHD    143.3304 5.780804 5.615446  28.85777      0.5892857 54.95536 27.04268
##         alcohol     age
## No_CHD 15.08656 38.8066
## CHD    17.01929 50.5000
```

```
# Valutazione performance QDA (train set)
qda_pred_train <- predict(qda_model, heart_train)
confusionMatrix(qda_pred_train$class, heart_train$y)
```

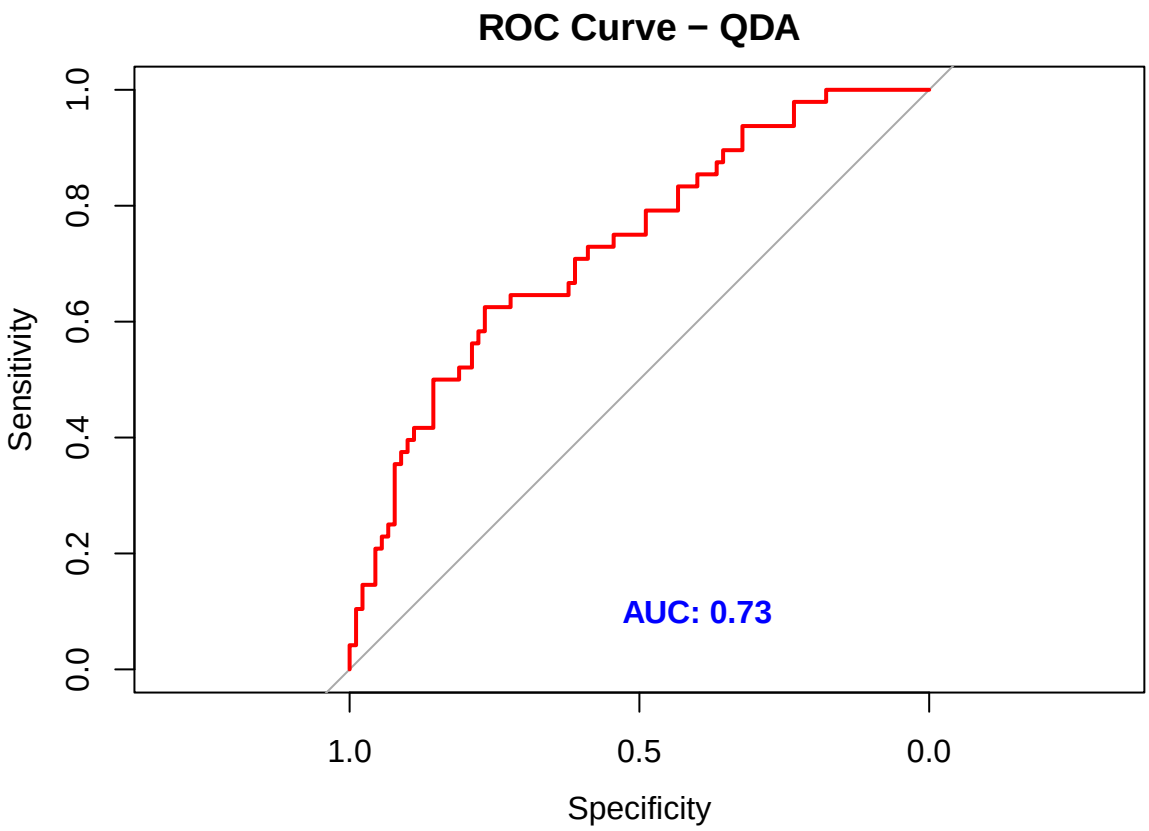
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No_CHD CHD
##     No_CHD    181  45
##     CHD        31  67
##
##                Accuracy : 0.7654
##                  95% CI : (0.7154, 0.8105)
##     No Information Rate : 0.6543
##     P-Value [Acc > NIR] : 9.685e-06
##
##                   Kappa : 0.4657
##
##  Mcnemar's Test P-Value : 0.1359
##
##             Sensitivity : 0.8538
##             Specificity : 0.5982
##          Pos Pred Value : 0.8009
##          Neg Pred Value : 0.6837
##              Prevalence : 0.6543
##          Detection Rate : 0.5586
##    Detection Prevalence : 0.6975
##       Balanced Accuracy : 0.7260
##
```

```
##        'Positive' Class : No_CHD
##
```

```
#Valutazione performance QDA (test set)
qda_pred_test <- predict(qda_model, heart_test)
confusionMatrix(qda_pred_test$class, heart_test$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction No_CHD CHD
##     No_CHD     73  24
##     CHD        17  24
##
##                Accuracy : 0.7029
##                  95% CI : (0.6192, 0.7776)
##     No Information Rate : 0.6522
##     P-Value [Acc > NIR] : 0.1219
##
##                   Kappa : 0.3221
##
##  Mcnemar's Test P-Value : 0.3487
##
##             Sensitivity : 0.8111
##             Specificity : 0.5000
##          Pos Pred Value : 0.7526
##          Neg Pred Value : 0.5854
##              Prevalence : 0.6522
##          Detection Rate : 0.5290
##    Detection Prevalence : 0.7029
##       Balanced Accuracy : 0.6556
##
##        'Positive' Class : No_CHD
##
```

```
#ROC curve e AUC per QDA
roc_qda <- roc(heart_test$y, qda_pred_test$posterior[,2])
auc_value= auc(roc_qda)
plot(roc_qda, main="ROC Curve - QDA", col="red")
text(0.4, 0.1, paste("AUC:", round(auc_value, 3)), col = "blue", font = 2)
```



Un valore di AUC di circa 0.7303 è considerato discreto, suggerendo che il modello QDA ha una buona capacità discriminante, anche se non ottimale. Confrontando questo risultato con quello di LDA (AUC = 0.7495), possiamo dire che: LDA e QDA mostrano prestazioni abbastanza simili, con un leggero vantaggio per LDA (0.7495). Tuttavia, dato che LDA viola significativamente l'assunzione delle covarianze, i risultati della QDA appaiono più affidabili.

Alla luce delle violazioni riscontrate, e nonostante la buona performance discriminante osservata, è prudente considerare la regressione logistica o tecniche non-parametriche come Random Forest o SVM.

```
data(heart)
heart <- data.frame(heart)

# definiamo il modello completo ed il modello nullo
mod_null <- glm(y ~ 1, family = "binomial", data = heart_numeric)
mod_full <- glm(y ~ ., family = "binomial", data = heart_numeric)

library(car)
ll_full <- logLik(mod_full)  # Log-likelihood del modello completo
ll_null <- logLik(update(mod_full, . ~ 1))  # Log-likelihood del modello nullo
pseudo_r2_mod <- 1 - (as.numeric(ll_full) / as.numeric(ll_null))
summary(mod_full)
```

```
##
## Call:
## glm(formula = y ~ ., family = "binomial", data = heart_numeric)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
```

```
## (Intercept) -6.066864   1.271443  -4.772 1.83e-06 ***
## sbp          0.005641   0.005611   1.005 0.314721
## tobacco      0.072716   0.026326   2.762 0.005742 **
## ldl          0.192492   0.059429   3.239 0.001199 **
## adiposity    0.017066   0.028433   0.600 0.548355
## typea        0.040467   0.012078   3.350 0.000807 ***
## obesity     -0.057931   0.042980  -1.348 0.177703
## alcohol      0.001446   0.004403   0.328 0.742627
## age          0.050650   0.011766   4.305 1.67e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 488.89  on 453  degrees of freedom
## AIC: 506.89
##
## Number of Fisher Scoring iterations: 4
```

```
Anova(mod_full)
```

```
## Analysis of Deviance Table (Type II tests)
##
## Response: y
##           LR Chisq Df Pr(>Chisq)
## sbp         1.0162  1  0.3134108
## tobacco     8.1944  1  0.0042021 **
## ldl        11.1180  1  0.0008549 ***
## adiposity   0.3614  1  0.5477539
## typea      11.9145  1  0.0005570 ***
## obesity     1.8621  1  0.1723804
## alcohol     0.1074  1  0.7431019
## age        19.5337  1  9.884e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
library(MASS)  # Per stepAIC()
library(dplyr)

# Selezione backward
mod1 <- stepAIC(mod_full, direction="backward")
```

```
## Start:  AIC=506.89
## y ~ sbp + tobacco + ldl + adiposity + typea + obesity + alcohol +
##     age
##
##             Df Deviance    AIC
## - alcohol    1   488.99 504.99
## - adiposity  1   489.25 505.25
## - sbp        1   489.90 505.90
## - obesity    1   490.75 506.75
## <none>           488.89 506.89
## - tobacco    1   497.08 513.08
## - ldl        1   500.00 516.00
```

```
## - typea      1   500.80 516.80
## - age        1   508.42 524.42
##
## Step:  AIC=504.99
## y ~ sbp + tobacco + ldl + adiposity + typea + obesity + age
##
##             Df Deviance    AIC
## - adiposity  1   489.38 503.38
## - sbp        1   490.11 504.11
## - obesity    1   490.90 504.90
## <none>           488.99 504.99
## - tobacco    1   497.86 511.86
## - ldl        1   500.00 514.00
## - typea      1   500.97 514.97
## - age        1   508.42 522.42
##
## Step:  AIC=503.38
## y ~ sbp + tobacco + ldl + typea + obesity + age
##
##           Df Deviance    AIC
## - sbp      1   490.58 502.58
## - obesity  1   491.24 503.24
## <none>         489.38 503.38
## - tobacco  1   498.35 510.35
## - typea    1   501.07 513.07
## - ldl      1   501.94 513.94
## - age      1   519.00 531.00
##
## Step:  AIC=502.58
## y ~ tobacco + ldl + typea + obesity + age
##
##           Df Deviance    AIC
## - obesity  1   492.09 502.09
## <none>         490.58 502.58
## - tobacco  1   499.72 509.72
## - typea    1   501.93 511.93
## - ldl      1   503.23 513.23
## - age      1   526.29 536.29
##
## Step:  AIC=502.09
## y ~ tobacco + ldl + typea + age
##
##           Df Deviance    AIC
## <none>         492.09 502.09
## - tobacco  1   501.27 509.27
## - typea    1   502.82 510.82
## - ldl      1   503.30 511.30
## - age      1   526.37 534.37
```

```
summary(mod1)
```

```
##
## Call:
## glm(formula = y ~ tobacco + ldl + typea + age, family = "binomial",
##     data = heart_numeric)
```

```
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.334452   0.897809  -7.055 1.72e-12 ***
## tobacco      0.075031   0.025699   2.920  0.00350 **
## ldl          0.179891   0.055027   3.269  0.00108 **
## typea        0.037914   0.011885   3.190  0.00142 **
## age          0.055040   0.009948   5.533 3.15e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 492.09  on 457  degrees of freedom
## AIC: 502.09
##
## Number of Fisher Scoring iterations: 4
```

```
# Selezione forward
mod2 <- stepAIC(mod_null, scope=list(lower=mod_null,upper=mod_full),
                direction="forward")
```

```
## Start:  AIC=598.11
## y ~ 1
##
##             Df Deviance    AIC
## + age        1   525.56 529.56
## + tobacco    1   554.65 558.65
## + ldl        1   564.28 568.28
## + adiposity  1   565.05 569.05
## + sbp        1   579.32 583.32
## + typea      1   591.12 595.12
## + obesity    1   591.53 595.53
## <none>           596.11 598.11
## + alcohol    1   594.35 598.35
##
## Step:  AIC=529.56
## y ~ age
##
##             Df Deviance    AIC
## + ldl        1   512.48 518.48
## + typea      1   513.04 519.04
## + tobacco    1   515.39 521.39
## <none>           525.56 529.56
## + sbp        1   524.44 530.44
## + adiposity  1   524.78 530.78
## + alcohol    1   524.89 530.89
## + obesity    1   525.55 531.55
##
## Step:  AIC=518.48
## y ~ age + ldl
##
##             Df Deviance    AIC
## + typea      1   501.27 509.27
```

```
## + tobacco    1   502.82 510.82
## <none>           512.48 518.48
## + alcohol    1   511.23 519.23
## + obesity    1   511.62 519.62
## + sbp        1   511.67 519.67
## + adiposity  1   512.34 520.34
##
## Step:  AIC=509.27
## y ~ age + ldl + typea
##
##             Df Deviance    AIC
## + tobacco    1   492.09 502.09
## <none>           501.27 509.27
## + obesity    1   499.72 509.72
## + alcohol    1   500.20 510.20
## + sbp        1   500.31 510.31
## + adiposity  1   501.10 511.10
##
## Step:  AIC=502.09
## y ~ age + ldl + typea + tobacco
##
##             Df Deviance    AIC
## <none>           492.09 502.09
## + obesity    1   490.58 502.58
## + sbp        1   491.24 503.24
## + alcohol    1   491.87 503.87
## + adiposity  1   491.89 503.89
```

```
summary(mod2)
```

```
##
## Call:
## glm(formula = y ~ age + ldl + typea + tobacco, family = "binomial",
##     data = heart_numeric)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -6.334452   0.897809  -7.055 1.72e-12 ***
## age          0.055040   0.009948   5.533 3.15e-08 ***
## ldl          0.179891   0.055027   3.269  0.00108 **
## typea        0.037914   0.011885   3.190  0.00142 **
## tobacco      0.075031   0.025699   2.920  0.00350 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 596.11  on 461  degrees of freedom
## Residual deviance: 492.09  on 457  degrees of freedom
## AIC: 502.09
##
## Number of Fisher Scoring iterations: 4
```

```
# Confronto AIC tra i modelli
aic_values <- AIC(mod_full, mod1, mod2)
```

```r
bic_values <- BIC(mod_full, mod1, mod2)

# Creazione di un dataframe per il confronto
comparison_table <- data.frame(
Modello = rownames(aic_values),  # Nomi dei modelli
AIC = aic_values$AIC,            # Valori AIC
BIC = bic_values$BIC             # Valori BIC
)
kable(comparison_table, caption = "Confronto tra Modelli: AIC e BIC")
```

Table 4: Confronto tra Modelli: AIC e BIC

Modello	AIC	BIC
mod_full	506.8851	544.1052
mod1	502.0948	522.7727
mod2	502.0948	522.7727

```r
library(pROC)

# Fare previsioni con il modello `mod1`
pred_prob <- predict(mod1, type = "response")

# Creare la curva ROC
roc_curve <- roc(heart$y, pred_prob)

# Calcolare lo Youden Index per trovare la soglia ottimale
youden_index <- roc_curve$sensitivities + roc_curve$specificities - 1
optimal_threshold <- roc_curve$thresholds[which.max(youden_index)]

# Stampare la soglia ottimale
cat("Soglia Ottimale (Youden Index):", optimal_threshold, "\n")
```

```
## Soglia Ottimale (Youden Index): 0.4025309
```

```r
#  Convertire le probabilità in classi usando la soglia ottimale
pred_class_opt <- ifelse(pred_prob > optimal_threshold, 1, 0)

#  Creare la matrice di confusione con la soglia ottimale
conf_matrix_opt <- confusionMatrix(as.factor(pred_class_opt), as.factor(heart$y))

# Stampare i risultati
print(conf_matrix_opt)
```

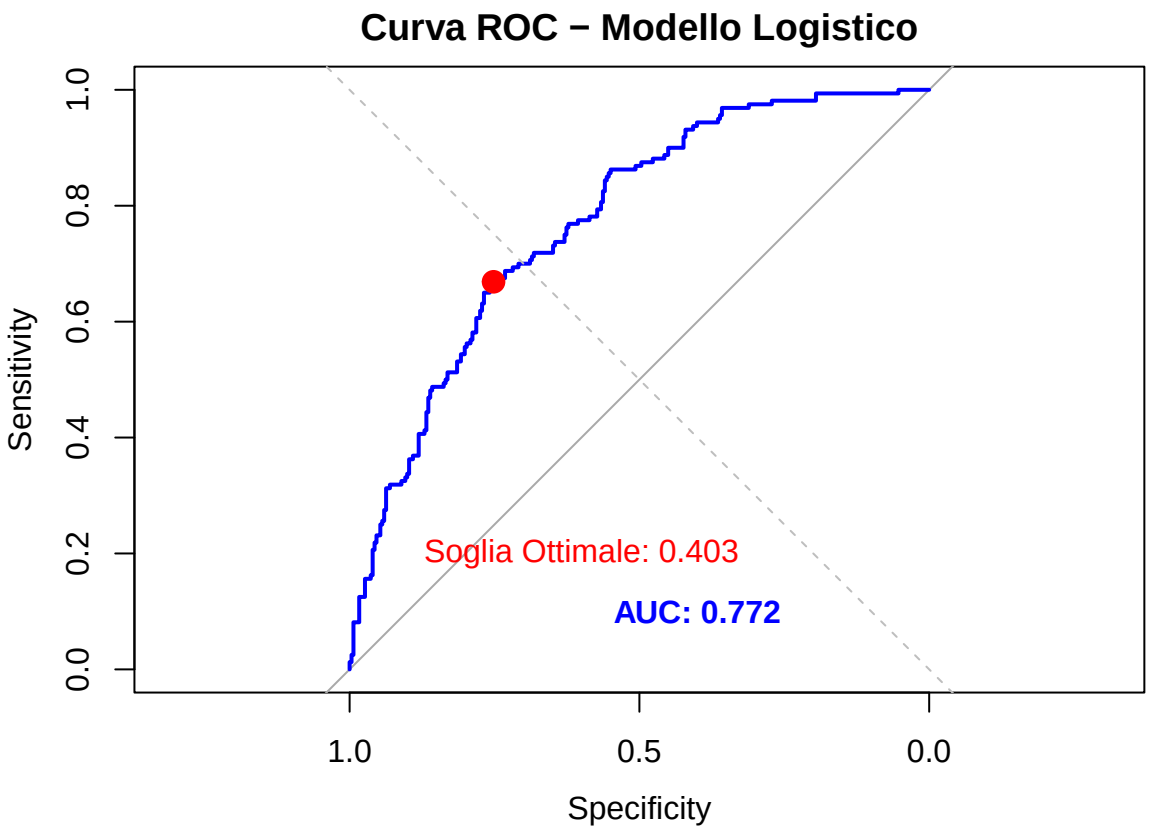
```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction   0   1
##          0 227  53
##          1  75 107
##
##                Accuracy : 0.7229
##                  95% CI : (0.6797, 0.7633)
##     No Information Rate : 0.6537
```

```
##     P-Value [Acc > NIR] : 0.0008747
##
##                   Kappa : 0.4072
##
##  Mcnemar's Test P-Value : 0.0634314
##
##             Sensitivity : 0.7517
##             Specificity : 0.6687
##          Pos Pred Value : 0.8107
##          Neg Pred Value : 0.5879
##              Prevalence : 0.6537
##          Detection Rate : 0.4913
##    Detection Prevalence : 0.6061
##       Balanced Accuracy : 0.7102
##
##        'Positive' Class : 0
##
```

```
auc_value <- auc(roc_curve)
cat("AUC:", auc_value, "\n")
```

```
## AUC: 0.7719164
```

```
#  Visualizzare la curva ROC
plot(roc_curve, col = "blue", main = "Curva ROC - Modello Logistico")
abline(a = 0, b = 1, lty = 2, col = "gray")  # Linea di riferimento casuale
points(roc_curve$specificities[which.max(youden_index)],
       roc_curve$sensitivities[which.max(youden_index)],
       col = "red", pch = 19, cex = 1.5)# Punto corrispondente alla soglia ottimale
text(0.6, 0.2, paste("Soglia Ottimale:", round(optimal_threshold, 3)), col = "red")
text(0.4, 0.1, paste("AUC:", round(auc_value, 3)), col = "blue", font = 2)
```



Il modello presentato è un modello di regressione logistica finalizzato a spiegare la probabilità dell'evento y = 1 a partire da alcune covariate (sbp, tobacco, ldl, adiposity, typea, obesity, alcohol, age). Nel codice, dopo aver definito un modello nullo – contenente solo l'intercetta – e un modello completo – che include tutte le variabili disponibili –, si procede a un'analisi per passi (stepwise) in cui si cerca la combinazione di regressori più adatta a bilanciare capacità esplicativa e parsimonia, in base a criteri come l'AIC e il BIC. Sia nell'approccio backward, partendo dal modello completo e rimuovendo le variabili meno significative, sia con l'approccio forward, partendo dal modello nullo e aggiungendo di volta in volta le variabili più promettenti, si converge a un sottoinsieme di quattro covariate: tobacco, ldl, typea e age.

A livello interpretativo, questo significa che, secondo i dati analizzati, l'uso di tabacco (tobacco), il livello di colesterolo LDL (ldl), il temperamento di tipo A (typea) e l'età (age) esercitano effetti statisticamente significativi sulla probabilità di incorrere nell'evento di interesse, al punto da rendere superfluo – per fini predittivi – l'inserimento di sbp, adiposity, obesity e alcohol. Il modello "definitivo" è dunque un logit in cui la probabilità di y=1 si esprime come funzione lineare di tobacco, ldl, typea e age. Tali variabili presentano coefficienti positivi, suggerendo che un aumento di ciascuna di esse incrementa il log-odds (e dunque la probabilità) dell'esito. L'intercetta negativa, invece, riflette la bassa probabilità dell'evento quando gli altri regressori sono ai valori minimi.

Per valutare la bontà di questo modello, dopo avere stimato i coefficienti via massima verosimiglianza, si è calcolata la curva ROC sul dataset, ottenendo un'area sotto la curva (AUC) di circa 0.77. Si tratta di un valore generalmente considerato discreto, in quanto indica un'abilità di discriminazione superiore al caso casuale (AUC=0.5) ma non ancora eccellente (che si avrebbe se AUC fosse vicina a 1). È poi stato applicato lo Youden Index per individuare una soglia di classificazione ottimale diversa da 0.5; la soglia risultante, intorno a 0.40, massimizza la somma di sensibilità e specificità. Questo vuol dire che, se l'algoritmo produce una probabilità di y=1 superiore a 0.40, conviene etichettare l'osservazione come "1" per migliorare nel complesso la capacità di classificazione.

La matrice di confusione risultante, calcolata con questa soglia, mostra un'accuratezza intorno al 72%.

Il modello, quindi, classifica correttamente circa sette individui su dieci. Più in dettaglio, la sensibilità (proporzione di veri positivi sul totale di positivi reali) raggiunge circa il 75%, mentre la specificità (proporzione di veri negativi sul totale di negativi reali) si aggira intorno al 67%. Questa matrice mette in evidenza come il modello, con la soglia scelta, tenda leggermente a predire l'uscita "0" piuttosto che "1", pur catturando comunque un numero discreto di casi positivi. In ogni caso, l'insieme di sensibilità, specificità, accuratezza e AUC indica che il modello logistic-stepwise con le quattro variabili selezionate è sufficientemente robusto e flessibile nel discriminare tra i due gruppi y=0 e y=1.

Dal punto di vista pratico, l'utilità del modello sta nella possibilità di calcolare, per ciascun individuo, una probabilità predetta di esito positivo in base ai valori di tobacco, ldl, typea e age: l'alzarsi di uno o più di questi fattori fa crescere la probabilità. Questo tipo di informazione consente di identificare profili più o meno esposti e può guidare sia interventi di prevenzione sia strategie di screening mirato.

```r
# Caricamento delle librerie necessarie
library(flexmix)
library(caret)
set.seed(123)

# Creazione del dataframe
df <- heart[, c("age", "obesity", "ldl", "sbp")]

# Definizione della variabile target
df$y <- as.numeric(heart$y)  # Convertiamo y in numerico

# Creazione della variabile binaria in formato matriciale (successi e fallimenti)
df$y_bin <- cbind(success = df$y, failure = 1 - df$y)

# Controllo che non ci siano righe con (0,0) o valori negativi
df <- df[rowSums(df$y_bin) > 0, ]

# Applicazione del modello GMM con FLXMRglm
mix_model1 <- stepFlexmix(y_bin ~ age+obesity+ldl+sbp,
                          data = df,
                          k = 2,
                          model = FLXMRglm(family = "binomial"),
                          control = list(minprior = 0.001, iter.max = 1000, tol = 1e-6),
                          nrep = 5)
```

```
## 2 : *
##  *
##  *
##  *
##  *
```

```r
# Risultati del modello
summary(mix_model1)
```

```
##
## Call:
## stepFlexmix(y_bin ~ age + obesity + ldl + sbp, data = df, model = FLXMRglm(family = "binomial"),
##     control = list(minprior = 0.001, iter.max = 1000, tol = 1e-06),
##     k = 2, nrep = 5)
##
##        prior size post>0 ratio
```

```
## Comp.1  0.72  372    462 0.805
## Comp.2  0.28   90    293 0.307
##
## 'log Lik.' -243.4062 (df=11)
## AIC: 508.8125   BIC: 554.3037
```

```r
# Assegnazione dei cluster al dataset originale
Cluster1 <- clusters(mix_model1)

C1 <- heart_vars[Cluster1 == "1", ]
C2 <- heart_vars[Cluster1 == "2", ]

# Creazione della tabella con medie, deviazioni standard e dimensione del cluster
df_cluster <- data.frame(
  "Media Cluster 1" = apply(C1, 2, mean),  # Medie Cluster 1
  "Deviazione Std Cluster 1" = apply(C1, 2, sd),  # Deviazione Standard Cluster 1
  "N Cluster 1" = nrow(C1),  # Numero di osservazioni Cluster 1
  "Media Cluster 2" = apply(C2, 2, mean),  # Medie Cluster 2
  "Deviazione Std Cluster 2" = apply(C2, 2, sd),  # Deviazione Standard Cluster 2
  "N Cluster 2" = nrow(C2)  # Numero di osservazioni Cluster 2
)

library(knitr)
kable(df_cluster, caption = "Profili dei Cluster")
```

Table 5: Profili dei Cluster

	Media.Cluster.1	Deviazione.Std.Cluster.1	N.Cluster.1	Media.Cluster.2	Deviazione.Std.Cluster.2	N.Cluster.2
obesity	26.230726	4.149811	372	25.272778	4.408413	90
ldl	4.698118	2.025452	372	4.914778	2.252744	90
age	41.147849	15.037677	372	49.711111	10.155249	90
sbp	137.494624	19.057226	372	141.766667	25.440855	90

```r
# Matrice di confusione confrontando la classificazione GMM con y_true
confusion1 <- confusionMatrix(as.factor(Cluster1 ==2),
                              as.factor(y_true==1))

# Stampiamo le metriche di valutazione
print(confusion1)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   284   88
##      TRUE     18   72
##
##                Accuracy : 0.7706
##                  95% CI : (0.7295, 0.8081)
##     No Information Rate : 0.6537
##     P-Value [Acc > NIR] : 3.266e-08
##
##                   Kappa : 0.4352
##
```

```
##  Mcnemar's Test P-Value : 2.058e-11
##
##             Sensitivity : 0.9404
##             Specificity : 0.4500
##          Pos Pred Value : 0.7634
##          Neg Pred Value : 0.8000
##              Prevalence : 0.6537
##          Detection Rate : 0.6147
##    Detection Prevalence : 0.8052
##       Balanced Accuracy : 0.6952
##
##        'Positive' Class : FALSE
##
```

Profilo dei Cluster Basato su Modelli di Mistura di Regressione Logistica L'analisi di clustering basata su modelli di mistura di regressione logistica ha identificato due gruppi distinti all'interno del dataset, caratterizzati da differenze significative nei valori medi delle variabili obesità, livelli di LDL, età e pressione sanguigna sistolica (SBP). Il Cluster 1, che comprende 372 soggetti, presenta una media dell'indice di obesità di 26.23 ($\pm$ 4.15), livelli medi di LDL pari a 4.70 ($\pm$ 2.03), un'età media di 41.15 anni ($\pm$ 15.04) e una pressione sistolica media di 137.49 mmHg ($\pm$ 19.06). D'altra parte, il Cluster 2, costituito da 90 soggetti, mostra valori leggermente più bassi per l'indice di obesità (25.27 $\pm$ 4.41), livelli di LDL superiori (4.91 $\pm$ 2.25), un'età media maggiore (49.71 $\pm$ 10.16 anni) e una pressione sistolica più alta (141.77 $\pm$ 25.44 mmHg).

Questi risultati suggeriscono che il Cluster 2 potrebbe essere rappresentativo di una popolazione più anziana con valori mediamente più elevati di pressione arteriosa e LDL, parametri frequentemente associati a un maggiore rischio cardiovascolare.

L'accuratezza complessiva del modello è pari a 77.06% (IC 95%: 72.95% - 80.81%), indicando una buona capacità predittiva rispetto alla distribuzione dei dati. Il valore p del test di McNemar (2.058e-11) suggerisce una differenza significativa tra gli errori di classificazione, suggerendo che il modello potrebbe avere una tendenza nel classificare le classi in modo differente.

Per quanto riguarda le metriche di sensibilità e specificità:

La sensibilità (recall per la classe positiva) è elevata, pari a 94.04%, indicando che il modello è altamente efficace nel rilevare i casi negativi. Tuttavia, la specificità è relativamente bassa (45.00%), suggerendo che il modello ha difficoltà nel distinguere correttamente i casi positivi. I valori predittivi riflettono queste osservazioni:

Il valore predittivo positivo (PPV) è del 76.34%, il che significa che quando il modello predice un caso negativo, ha una probabilità del 76.34% di essere corretto. Il valore predittivo negativo (NPV) è dell'80.00%, suggerendo che il modello è relativamente affidabile nel prevedere correttamente i casi positivi. Infine, l'accuratezza bilanciata (Balanced Accuracy) del modello è 69.52%, un valore moderato che tiene conto dello squilibrio tra le due classi. L'indice di kappa di Cohen (0.4352) indica un livello moderato di accordo tra le previsioni del modello e la verità empirica.

Questi risultati evidenziano che il modello ha una buona capacità di individuare i casi negativi, ma potrebbe beneficiare di un'ottimizzazione della specificità, al fine di migliorare la capacità di individuare i casi positivi. L'uso di tecniche di bilanciamento delle classi o di una migliore selezione delle variabili potrebbe contribuire a migliorare le prestazioni globali del modello.

```r
# Caricamento delle librerie necessarie
library(flexmix)
library(caret)

# Impostiamo il seme per riproducibilità
set.seed(123)
```

```r
# Creazione del dataframe con variabili indipendenti e target binario
df <- heart[, c("age", "obesity", "ldl", "sbp")]

# Definizione della variabile target come numerica
df$y <- as.numeric(heart$y)  # Convertiamo y in numerico

# Creazione della variabile binaria in formato matriciale (successi e fallimenti)
df$y_bin <- cbind(success = df$y, failure = 1 - df$y)

# Rimuoviamo eventuali righe con (0,0) o valori negativi
df <- df[rowSums(df$y_bin) > 0, ]

#  Modello di Mixture di Regressioni Logistiche con Concomitanti
mix_model2 <- stepFlexmix(
  y_bin ~ .,              # Modello completo con tutte le covariate
  concomitant = FLXPmultinom(~ age + obesity + ldl + sbp),  # Covariate per la probabilità di cluster
  data = df,
  k = 2,                  # Numero di cluster
  model = FLXMRglm(family = "binomial"),
  control = list(minprior = 0.001, iter.max = 1000, tol = 1e-6),
  nrep = 5
)
```

```
## 2 : *
##  *
##  *
##  *
##  *
```

```r
# Risultati del modello
summary(mix_model2)
```

```
##
## Call:
## stepFlexmix(y_bin ~ ., concomitant = FLXPmultinom(~age + obesity +
##     ldl + sbp), data = df, model = FLXMRglm(family = "binomial"),
##     control = list(minprior = 0.001, iter.max = 1000, tol = 1e-06),
##     k = 2, nrep = 5)
##
##        prior size post>0 ratio
## Comp.1 0.493  201    462 0.435
## Comp.2 0.507  261    462 0.565
##
## 'log Lik.' -4.930206e-10 (df=17)
## AIC: 34   BIC: 104.3046
```

```r
# Assegnazione dei cluster al dataset originale
Cluster2 <- clusters(mix_model2)

C1 <- heart_vars[Cluster2 == "1", ]
C2 <- heart_vars[Cluster2 == "2", ]

# Creazione della tabella con medie, deviazioni standard e dimensione del cluster
```

```r
df_cluster <- data.frame(
  "Media Cluster 1" = apply(C1, 2, mean),  # Medie Cluster 1
  "Deviazione Std Cluster 1" = apply(C1, 2, sd),  # Deviazione Standard Cluster 1
  "N Cluster 1" = nrow(C1),  # Numero di osservazioni Cluster 1
  "Media Cluster 2" = apply(C2, 2, mean),  # Medie Cluster 2
  "Deviazione Std Cluster 2" = apply(C2, 2, sd),  # Deviazione Standard Cluster 2
  "N Cluster 2" = nrow(C2)  # Numero di osservazioni Cluster 2
)

library(knitr)
kable(df_cluster, caption = "Profili dei Cluster")
```

Table 6: Profili dei Cluster

	Media.Cluster.1	Deviazione.Std.Cluster.1	N.Cluster.1	Media.Cluster.2	Deviazione.Std.Cluster.2	N.Cluster.2
obesity	24.298607	3.378092	201	27.388352	4.305084	261
ldl	3.673682	1.310922	201	5.561763	2.173773	261
age	43.825871	15.363741	201	42.038314	13.980153	261
sbp	147.527363	22.555055	201	131.241379	15.417497	261

```r
# Matrice di confusione confrontando la classificazione GMM con y_true
confusion2 <- confusionMatrix(as.factor(Cluster2 ==1),
                              as.factor(y_true==1))

# Stampiamo le metriche di valutazione
print(confusion2)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction FALSE TRUE
##      FALSE   169   92
##      TRUE    133   68
##
##                Accuracy : 0.513
##                  95% CI : (0.4664, 0.5594)
##     No Information Rate : 0.6537
##     P-Value [Acc > NIR] : 1.000000
##
##                   Kappa : -0.0145
##
##  Mcnemar's Test P-Value : 0.007661
##
##             Sensitivity : 0.5596
##             Specificity : 0.4250
##          Pos Pred Value : 0.6475
##          Neg Pred Value : 0.3383
##              Prevalence : 0.6537
##          Detection Rate : 0.3658
##    Detection Prevalence : 0.5649
##       Balanced Accuracy : 0.4923
##
##        'Positive' Class : FALSE
```

##

I due cluster individuati mostrano differenze significative in alcune variabili chiave.

Il Cluster 1 è composto da 201 individui, con una media dell'indice di obesità pari a 24.30 (DS = 3.38), un valore di LDL medio di 3.67 (DS = 1.31), un'età media di 43.83 anni (DS = 15.36) e una pressione sistolica media di 147.53 mmHg (DS = 22.55). Il Cluster 2, con 261 individui, presenta valori medi di obesità più elevati (27.39, DS = 4.31), così come livelli di LDL più alti (5.56, DS = 2.17). L'età media è leggermente inferiore rispetto al Cluster 1 (42.04, DS = 13.98), cosi come la pressione sistolica che risulta inferiore (131.24, DS = 15.42).

Questi risultati suggeriscono che il Cluster 2 è caratterizzato da un profilo con livelli di LDL e obesità superiori rispetto al Cluster 1, mentre la pressione sistolica risulta inferiore. Il Cluster 1, invece, mostra valori medi più contenuti per obesità e LDL, ma una pressione arteriosa più elevata ed un'eta superiore.

La matrice di confusione e le metriche di valutazione del modello indicano un'accuratezza complessiva del 51,3%, con un intervallo di confidenza al 95% tra 46,6% e 55,9%, suggerendo che il modello non supera significativamente la probabilità casuale (No Information Rate = 65,37%). Il valore Kappa negativo (-0,0145) indica che il modello ha una performance quasi casuale e una scarsa concordanza con le etichette reali.

Dal punto di vista delle prestazioni sui due gruppi, la sensibilità (recall per la classe "False") è pari al 55,96%, indicando che il modello identifica correttamente circa il 56% dei casi negativi. Tuttavia, la specificità è solo del 42,5%, il che implica una bassa capacità di individuare correttamente i veri positivi. Il valore della Balanced Accuracy (49,23%) conferma che il modello è vicino a una classificazione casuale. Inoltre, il test di McNemar (p-value = 0,0077) suggerisce una differenza significativa tra le due classi, il che indica una distribuzione squilibrata degli errori.

Complessivamente, le prestazioni del modello risultano insoddisfacenti per una classificazione affidabile.