Exploring-hierarchical-clustering

Princewill

2025-11-05

Introduction The purpose of this project was to explore hierarchical clustering techniques on a synthetic dataset with known group labels. The analysis evaluates whether meaningful clusters exist, identifies the optimal number of clusters, and compares five linkage methods—Single, Complete, Average, Ward.D, and Ward.D2—using internal and external validation measures.

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.1     v stringr   1.5.2
## v ggplot2   4.0.0     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(cluster)        # silhouette
library(factoextra)     # viz: fviz_*, get_clust_tendency
```

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

```r
library(clustertend)    # Hopkins, VAT
```

```
## Package `clustertend` is deprecated.  Use package `hopkins` instead.
```

```r
library(dendextend)     # dendrogram comparison
```

```
##
## ---------------------
## Welcome to dendextend version 1.19.1
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
```

```
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
##
##
## Attaching package: 'dendextend'
##
## The following object is masked from 'package:stats':
##
##     cutree
```

```
set.seed(42)

 # Data: synthetic with ground truth (  3 centers,   3 features)
n_per_group <- 25
centers <- matrix(c(0,0,0,
                    6,5,0,
                    -5, 2, 3,
                    3,-4, 6), ncol = 3, byrow = TRUE)  # 4 groups, 3 features
sd_within <- 1.4

mk_blob <- function(mu, n, sd) sweep(matrix(rnorm(n*length(mu), 0, sd), ncol=length(mu)), 2, mu, "+")
X_list <- lapply(1:nrow(centers), function(i) mk_blob(centers[i,], n_per_group, sd_within))
X <- do.call(rbind, X_list)
y <- factor(rep(paste0("G",1:nrow(centers)), each = n_per_group))
colnames(X) <- c("student1","student2","student3")
df <- as_tibble(X) %>% mutate(label = y)
head(df)
```

```
## # A tibble: 6 x 4
##   student1 student2 student3 label
##      <dbl>    <dbl>    <dbl> <fct>
## 1    1.92    -0.603    0.451 G1
## 2   -0.791   -0.360   -1.10  G1
## 3    0.508   -2.47     2.21  G1
## 4    0.886    0.644    0.900 G1
## 5    0.566   -0.896    0.126 G1
## 6   -0.149    0.638    0.387 G1
```

```
str(df)
```

```
## tibble [100 x 4] (S3: tbl_df/tbl/data.frame)
##  $ student1: num [1:100] 1.919 -0.791 0.508 0.886 0.566 ...
##  $ student2: num [1:100] -0.603 -0.36 -2.468 0.644 -0.896 ...
##  $ student3: num [1:100] 0.451 -1.097 2.206 0.9 0.126 ...
##  $ label   : Factor w/ 4 levels "G1","G2","G3",..: 1 1 1 1 1 1 1 1 1 1 ...
```

```
df
```

```
## # A tibble: 100 x 4
##    student1 student2 student3 label
##       <dbl>    <dbl>    <dbl> <fct>
##  1   1.92     -0.603    0.451 G1
##  2  -0.791    -0.360   -1.10  G1
##  3   0.508    -2.47     2.21  G1
##  4   0.886     0.644    0.900 G1
##  5   0.566    -0.896    0.126 G1
##  6  -0.149     0.638    0.387 G1
##  7   2.12      0.987    0.951 G1
##  8  -0.133     1.45     0.126 G1
##  9   2.83     -0.852   -4.19  G1
## 10  -0.0878    0.707    0.399 G1
## # i 90 more rows
```

Data Preparation A synthetic dataset containing 100 observations was generated using four predefined centers and three numeric features (student1, student2, and student3). Each group represented a unique cluster label (G1–G4).

```
# Standardize only numeric feature columns, exclude the label
Xscaled <- scale(df[, !names(df) %in% "label"])

head(Xscaled)
```

```
##         student1    student2     student3
## [1,]  0.17889958 -0.32151099 -0.612326987
## [2,] -0.44286831 -0.25339432 -1.145960272
## [3,] -0.14483364 -0.84563800 -0.007251681
## [4,] -0.05819015  0.02873444 -0.457427388
## [5,] -0.13161876 -0.40391408 -0.724367843
## [6,] -0.29556617  0.02690675 -0.634224380
```

To ensure comparability among variables, all numeric columns were standardized before distance calculation.

```
# Hopkins correctly

set.seed(42)

tendency <- get_clust_tendency(Xscaled, n = 75, graph = TRUE)
```

```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
tendency$hopkins_stat
```

```
## [1] 0.7511941
```

The Hopkins statistic (0.751) was computed to assess clustering tendency. Although slightly high, the data exhibited moderate potential for cluster formation. A visual assessment using the VAT map confirmed partial structure, justifying further hierarchical clustering.

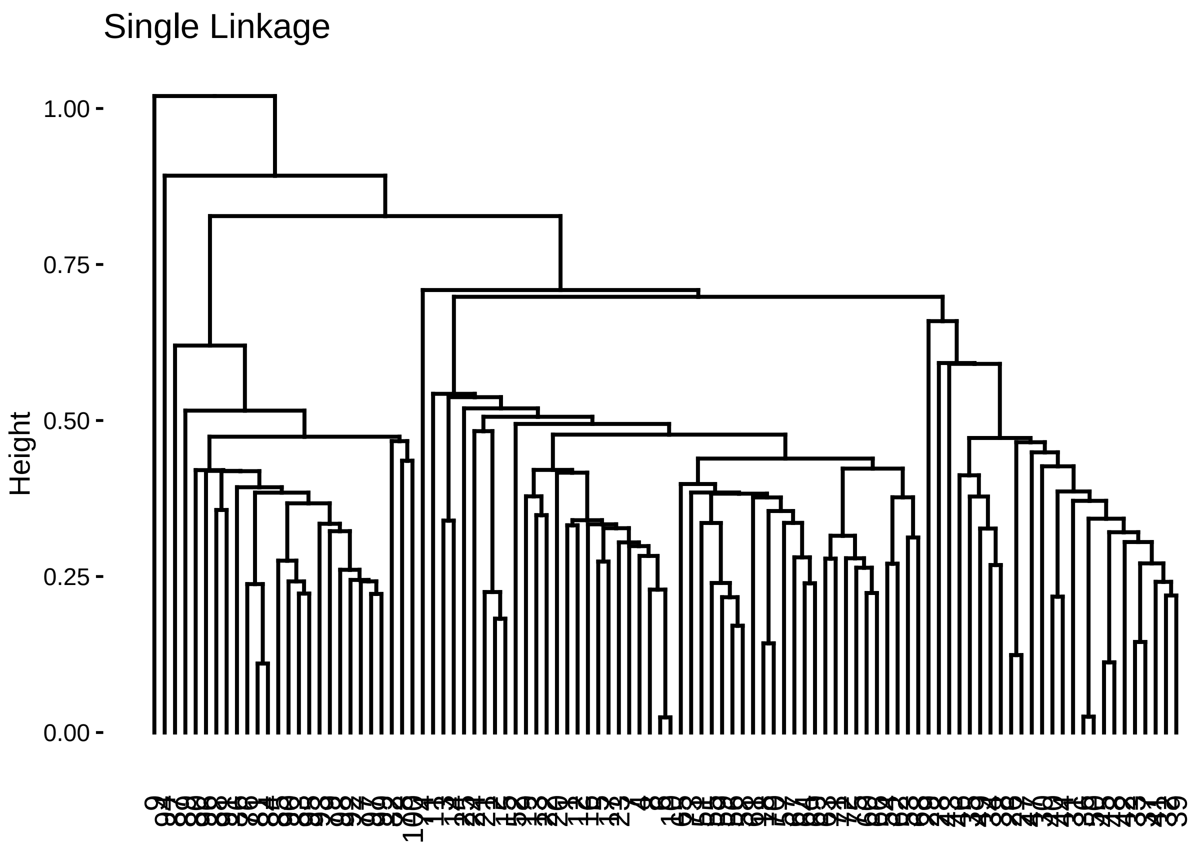
```
#Compute Distance Matrix
D <- dist(Xscaled, method = "euclidean")

#Apply All 5 Linkage Methods

hc_single   <- hclust(D, method = "single")

hc_complete <- hclust(D, method = "complete")

hc_average  <- hclust(D, method = "average")

hc_wardD    <- hclust(D, method = "ward.D")

hc_wardD2   <- hclust(D, method = "ward.D2")
```

Five linkage techniques were applied to the Euclidean distance matrix: • Single linkage (minimum distance) • Complete linkage (maximum distance) • Average linkage (mean distance) • Ward.D and Ward.D2 (variance-minimizing approaches) Each method was visualized using dendrograms and cluster plots to observe differences in grouping patterns. Ward's methods produced the most compact and interpretable clusters.

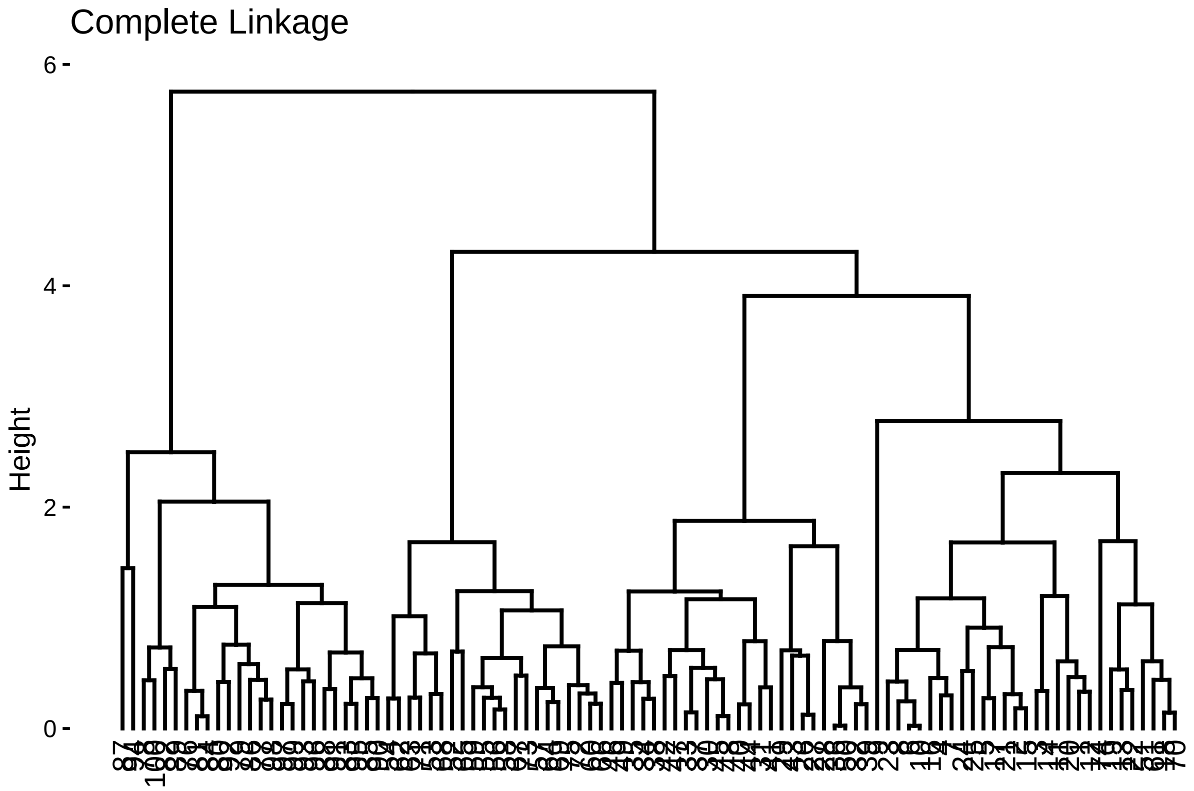
```
#Plot Dendrograms
fviz_dend(hc_single,   main = "Single Linkage")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

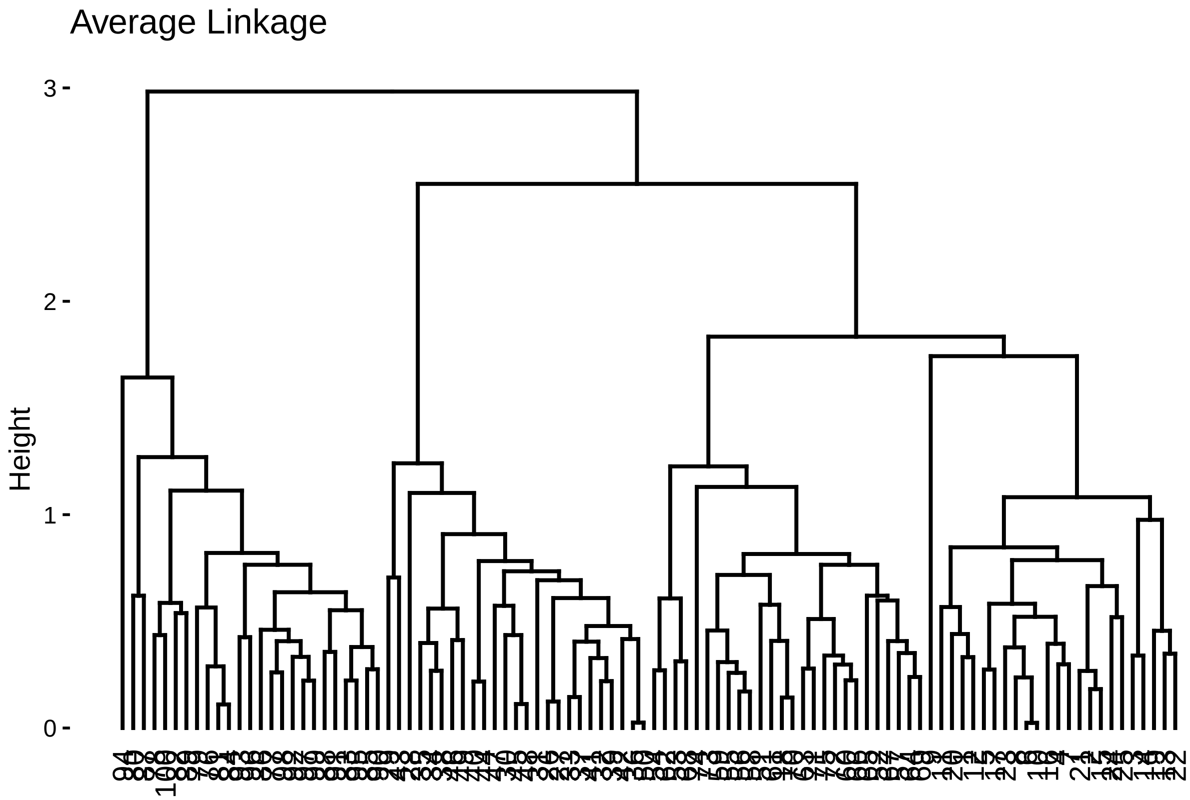
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
## of ggplot2 3.3.4.
## i The deprecated feature was likely used in the factoextra package.
##   Please report the issue at <https://github.com/kassambara/factoextra/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



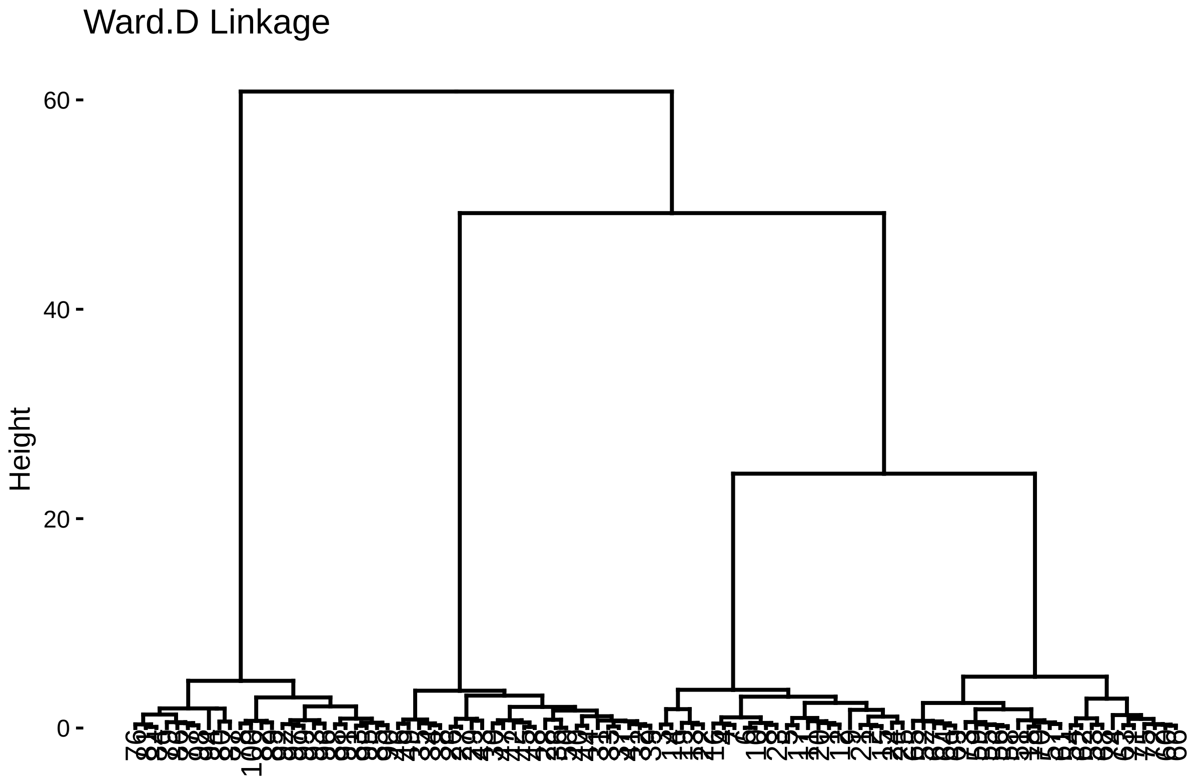
```r
fviz_dend(hc_complete, main = "Complete Linkage")
```



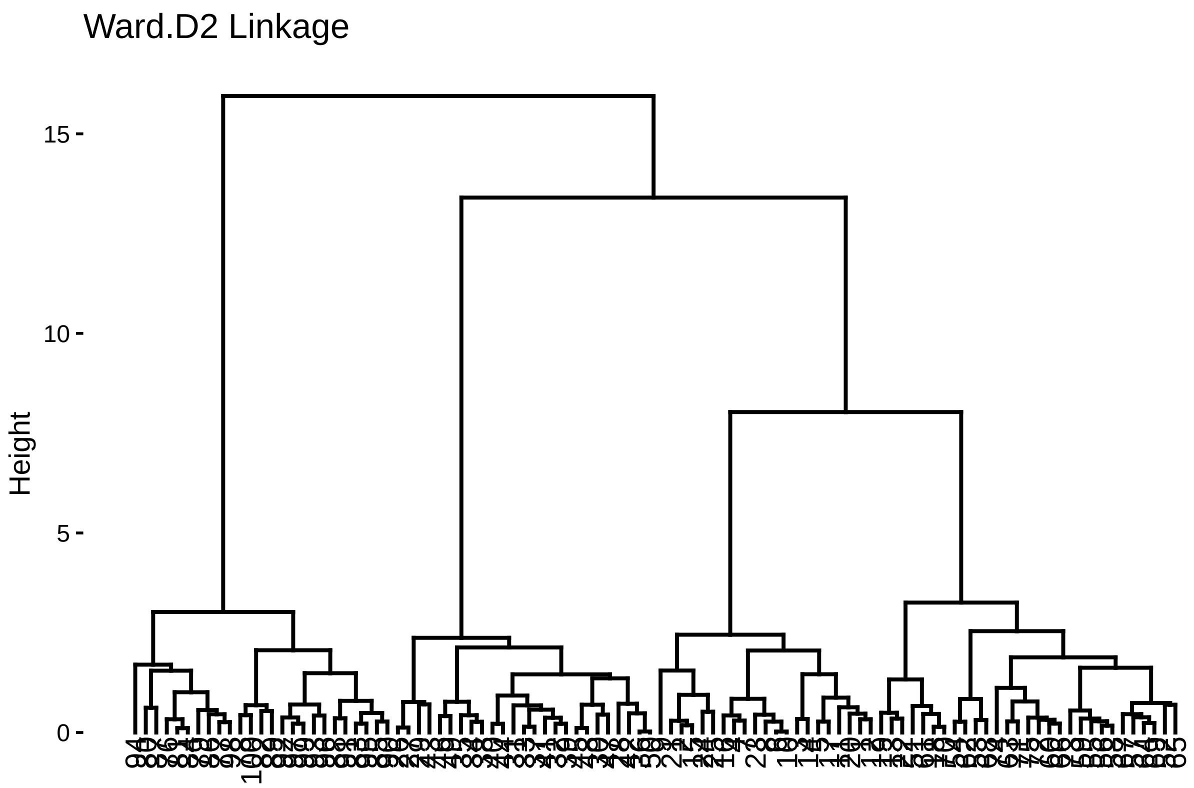
```r
fviz_dend(hc_average,  main = "Average Linkage")
```



```r
fviz_dend(hc_wardD,     main = "Ward.D Linkage")
```



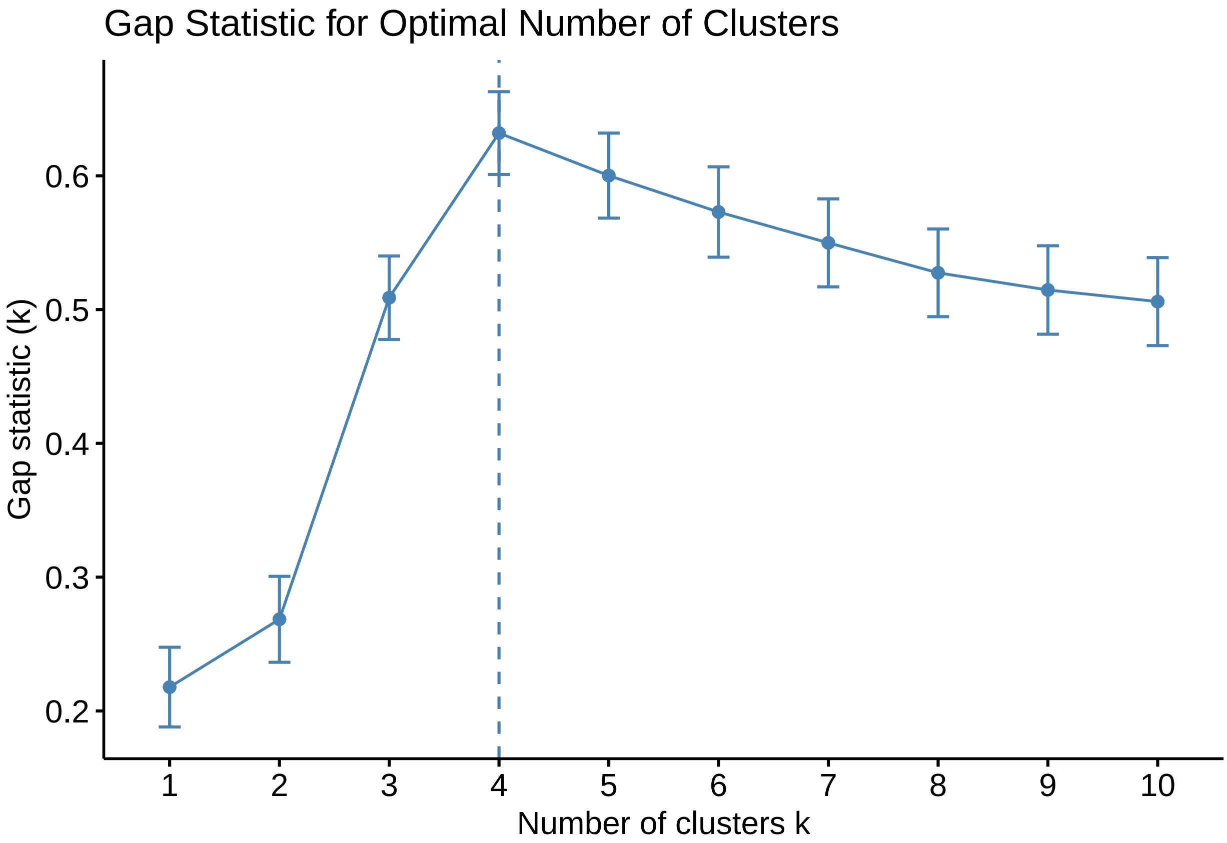
```r
fviz_dend(hc_wardD2,   main = "Ward.D2 Linkage")
```



```r
#Number of Clusters
set.seed(42)

# Compute Gap Statistic
gap_stat <- clusGap(
  Xscaled,                  # standardized numeric data
  FUN = hcut,               # hierarchical clustering wrapper for gap statistic
  K.max = 10,               # test from 1 to 10 clusters
  B = 100                   # number of bootstraps
)

# Print results
print(gap_stat)
```

```
## Clustering Gap statistic ["clusGap"] from call:
## clusGap(x = Xscaled, FUNcluster = hcut, K.max = 10, B = 100)
## B=100 simulated reference sets, k = 1..10; spaceH0="scaledPCA"
##  --> Number of clusters (method 'firstSEmax', SE.factor=1): 4
##          logW   E.logW       gap     SE.sim
##  [1,] 4.007039 4.224862 0.2178231 0.02977950
##  [2,] 3.683470 3.951956 0.2684859 0.03212345
##  [3,] 3.313741 3.822606 0.5088656 0.03122905
##  [4,] 3.082830 3.714753 0.6319231 0.03096664
##  [5,] 3.019979 3.620114 0.6001351 0.03177692
##  [6,] 2.963391 3.536324 0.5729331 0.03379312
##  [7,] 2.910727 3.460612 0.5498849 0.03287504
##  [8,] 2.864216 3.391682 0.5274661 0.03277134
##  [9,] 2.814171 3.328818 0.5146476 0.03307683
## [10,] 2.764426 3.270392 0.5059657 0.03289154
```

```
# Visualize the Gap Statistic
fviz_gap_stat(gap_stat) + ggtitle("Gap Statistic for Optimal Number of Clusters")
```



The Gap Statistic method (clusGap) identified 4 clusters as optimal. This agreed with the true number of groups in the simulated data and was confirmed by silhouette inspection.

```
#Cut Dendrograms & Compare to Ground Truth
set.seed(42)
k <- 4 # number of clusters to cut into
clusters_single   <- cutree(hc_single,   k = k)
clusters_complete <- cutree(hc_complete, k = k)
clusters_average  <- cutree(hc_average,  k = k)
clusters_wardD    <- cutree(hc_wardD,    k = k)
clusters_wardD2   <- cutree(hc_wardD2,   k = k)

table(clusters_single)
```

```
## clusters_single
##  1  2  3  4
## 74  1 24  1
```

```r
table(clusters_complete)
```

```
## clusters_complete
##  1  2  3  4
## 29 25 21 25
```

```r
table(clusters_average)
```

```
## clusters_average
##  1  2  3  4
## 24 26 25 25
```

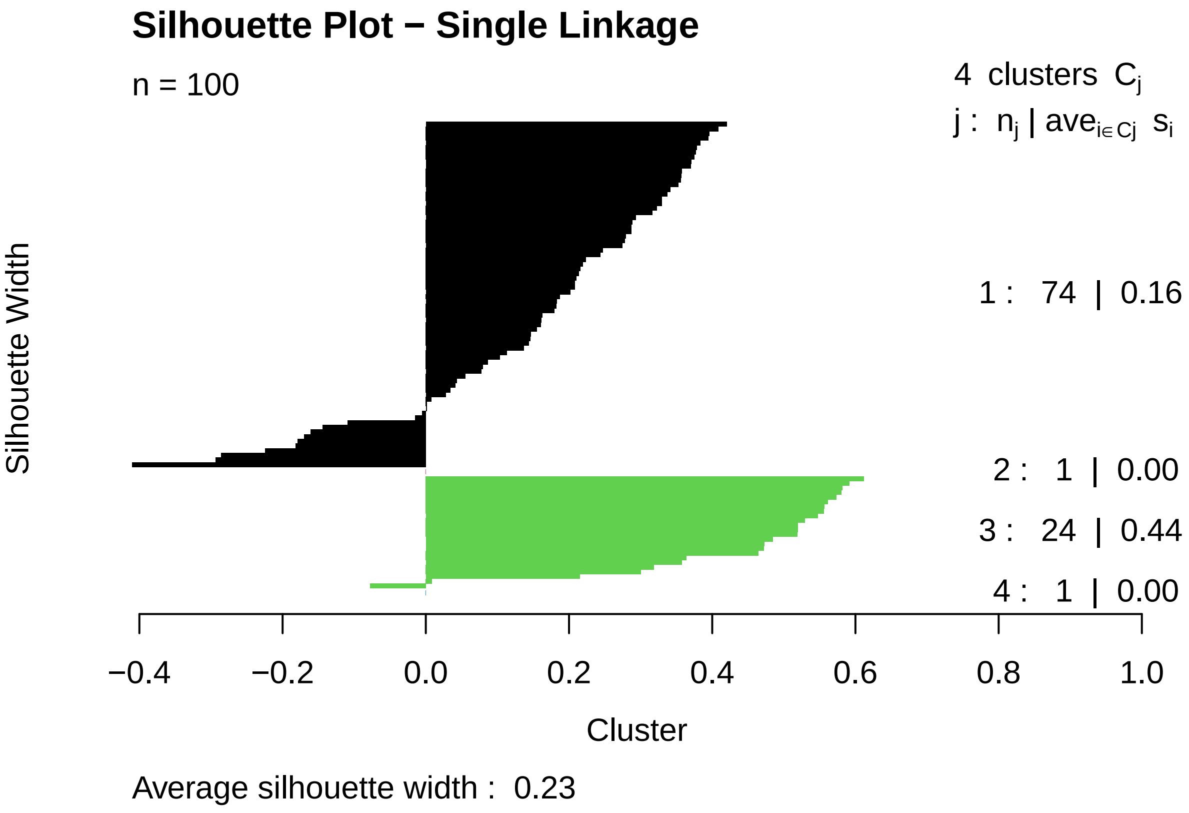
```r
table(clusters_wardD)
```

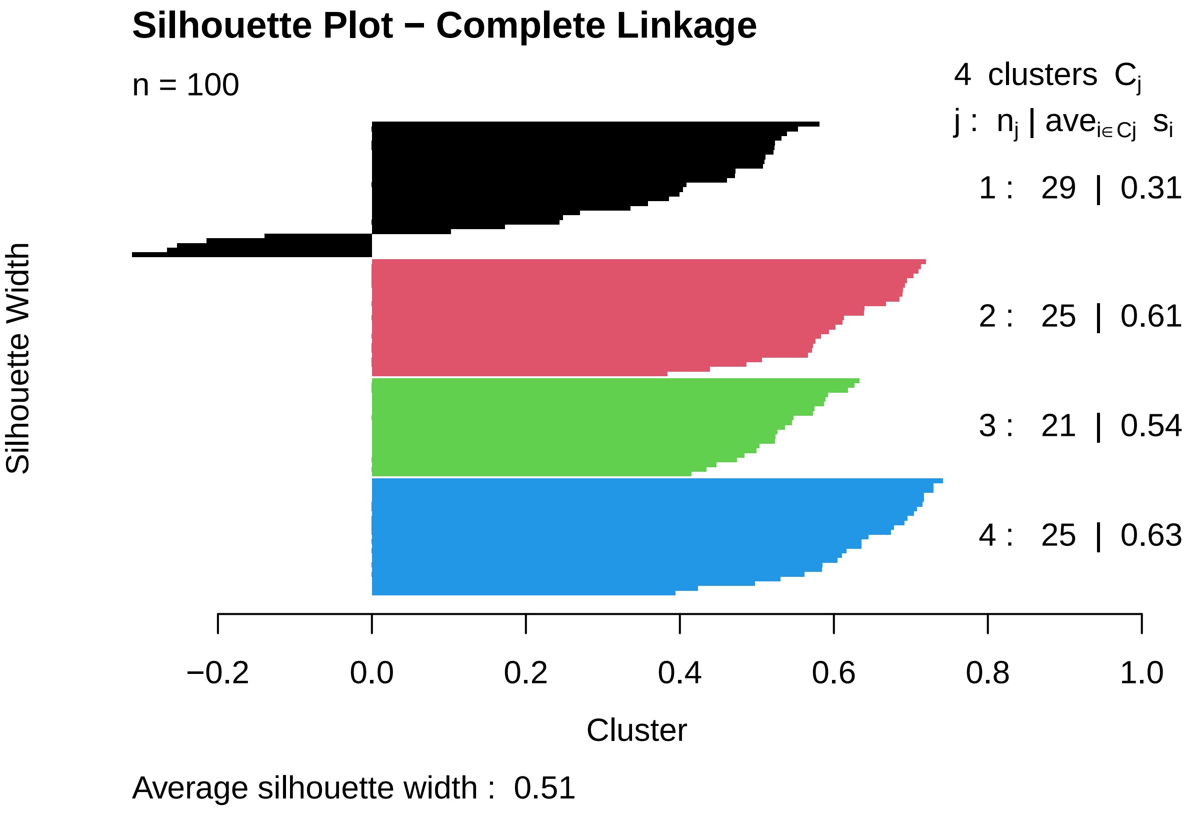
```
## clusters_wardD
##  1  2  3  4
## 24 26 25 25
```

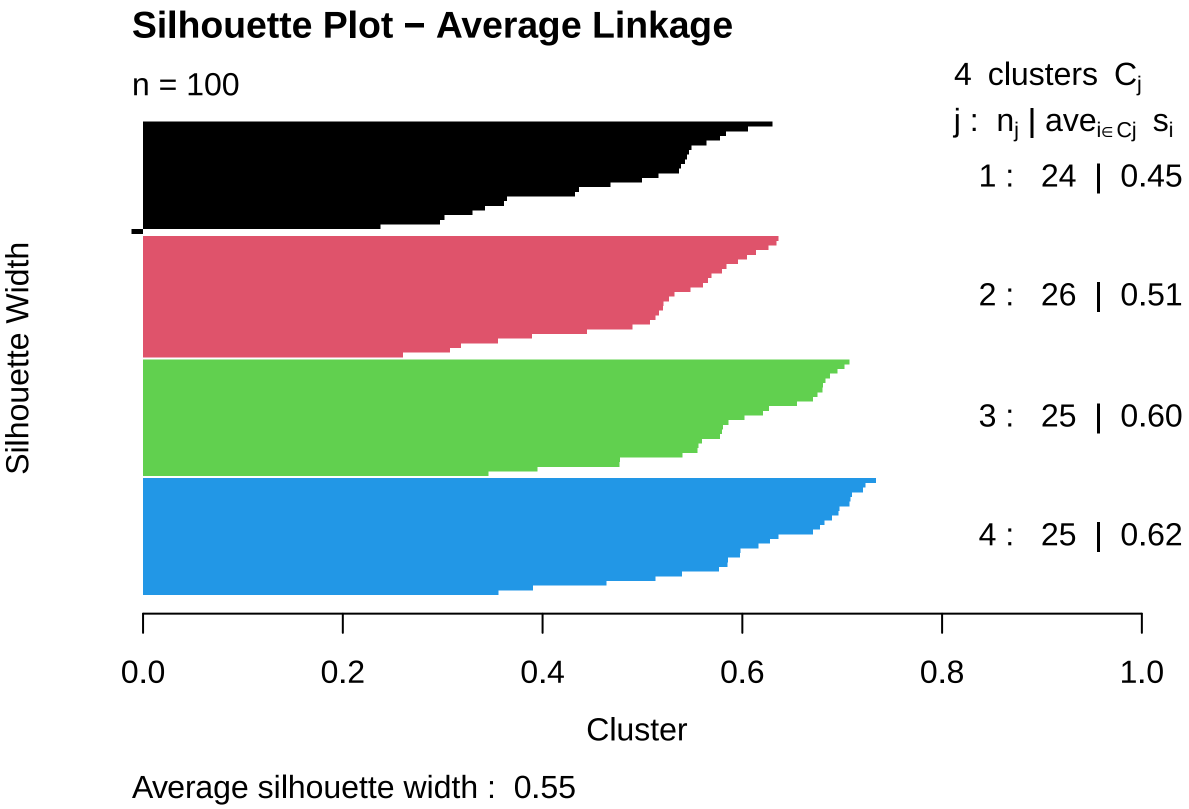
```r
table(clusters_wardD2)
```

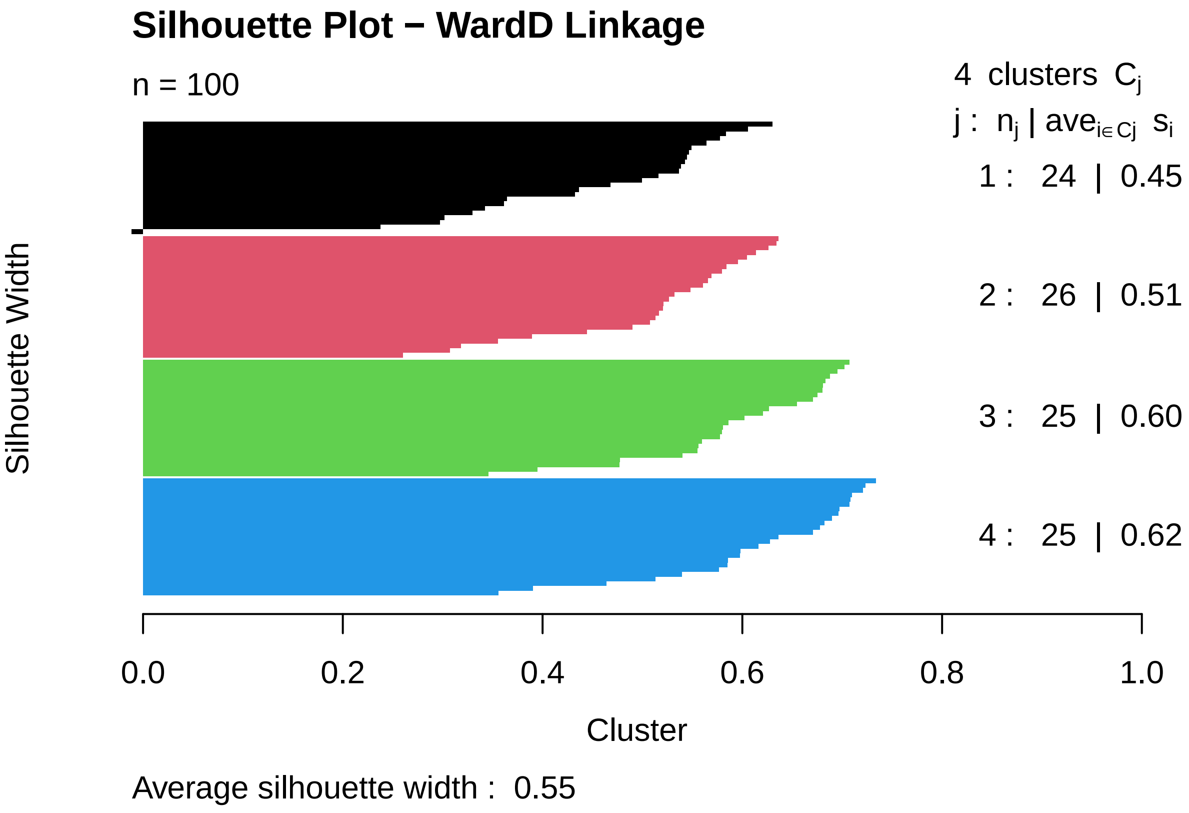
```
## clusters_wardD2
##  1  2  3  4
## 21 29 25 25
```

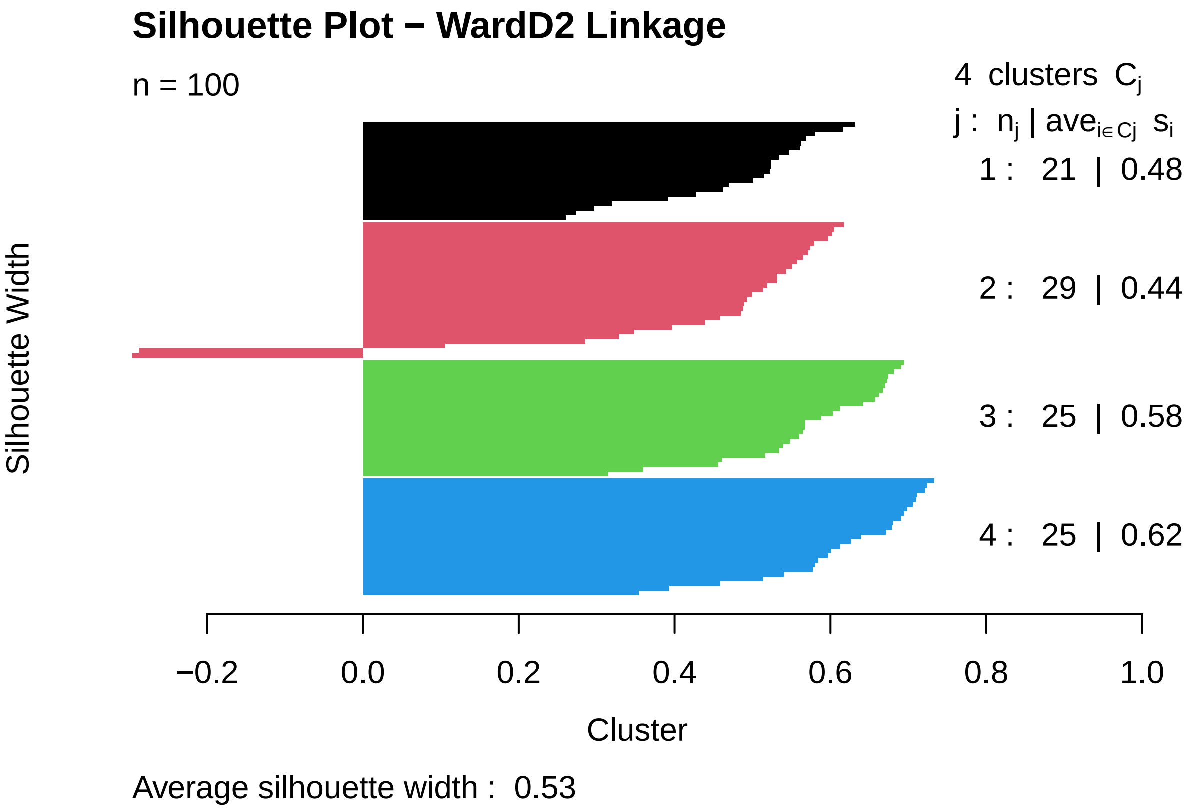
```r
#Silhouette Plots for All 5 Hierarchical Methods

# List of cluster assignments
methods <- list(
  Single   = clusters_single,
  Complete = clusters_complete,
  Average  = clusters_average,
  WardD    = clusters_wardD,
  WardD2   = clusters_wardD2
)

# Loop and generate silhouette graph for each
for (m in names(methods)) {
  sil <- silhouette(methods[[m]], D)
  plot(sil,
       main = paste("Silhouette Plot -", m, "Linkage"),
       xlab = "Cluster",
       ylab = "Silhouette Width",
       col = 1:max(methods[[m]]),
       border = NA)
}
```


Silhouette Plot − Single Linkage
n = 100
4 clusters C_j
$j :\ n_j \mid \text{ave}_{i \in C_j}\ s_i$
1 : 74 | 0.16
2 : 1 | 0.00
3 : 24 | 0.44
4 : 1 | 0.00
Silhouette Width
−0.4
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Cluster
Average silhouette width : 0.23

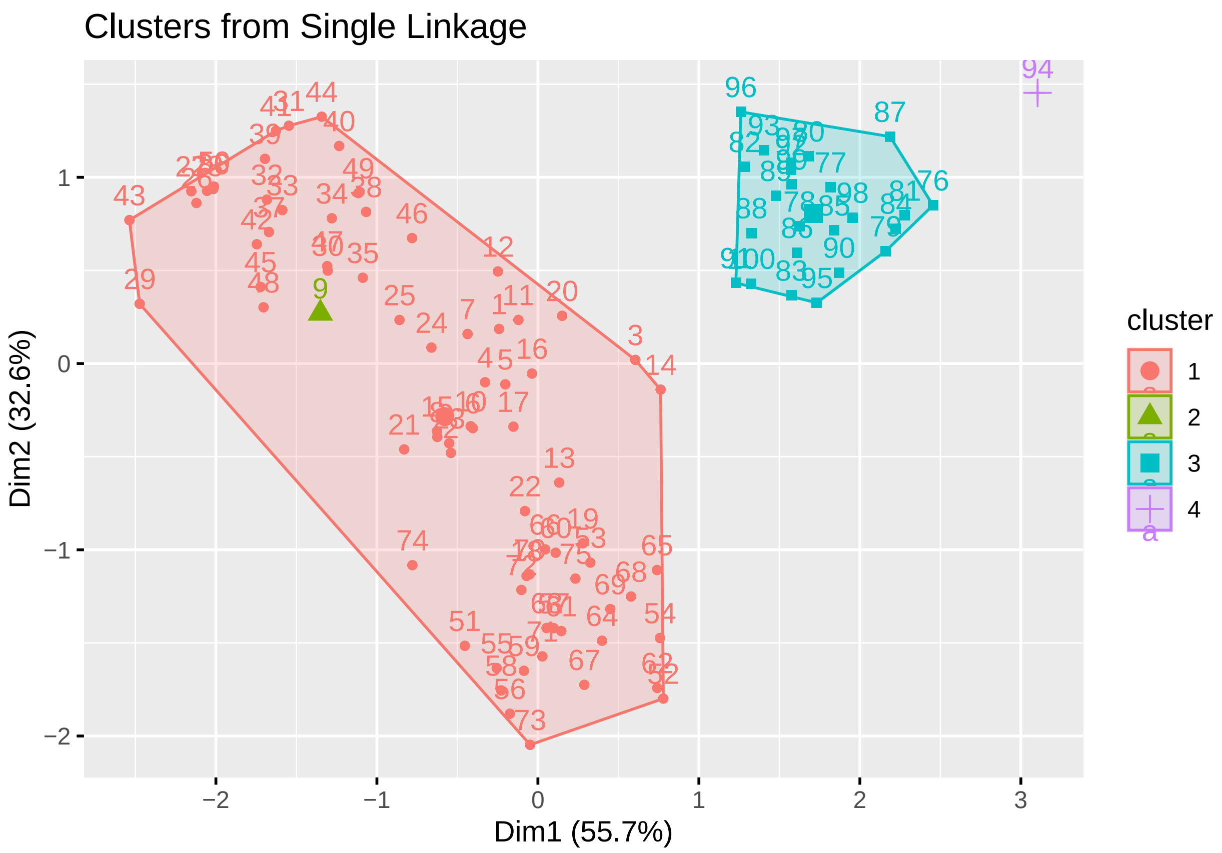

Silhouette Plot − Complete Linkage
n = 100
4 clusters C_j
j : n_j | $\text{ave}_{i \in Cj}$ s_i
1 : 29 | 0.31
2 : 25 | 0.61
3 : 21 | 0.54
4 : 25 | 0.63
Silhouette Width
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
Cluster
Average silhouette width : 0.51


Silhouette Plot − Average Linkage
n = 100
4 clusters C_j
j : n_j | $\text{ave}_{i \in C_j}$ s_i
1 : 24 | 0.45
2 : 26 | 0.51
3 : 25 | 0.60
4 : 25 | 0.62
Silhouette Width
0.0
0.2
0.4
0.6
0.8
1.0
Cluster
Average silhouette width : 0.55

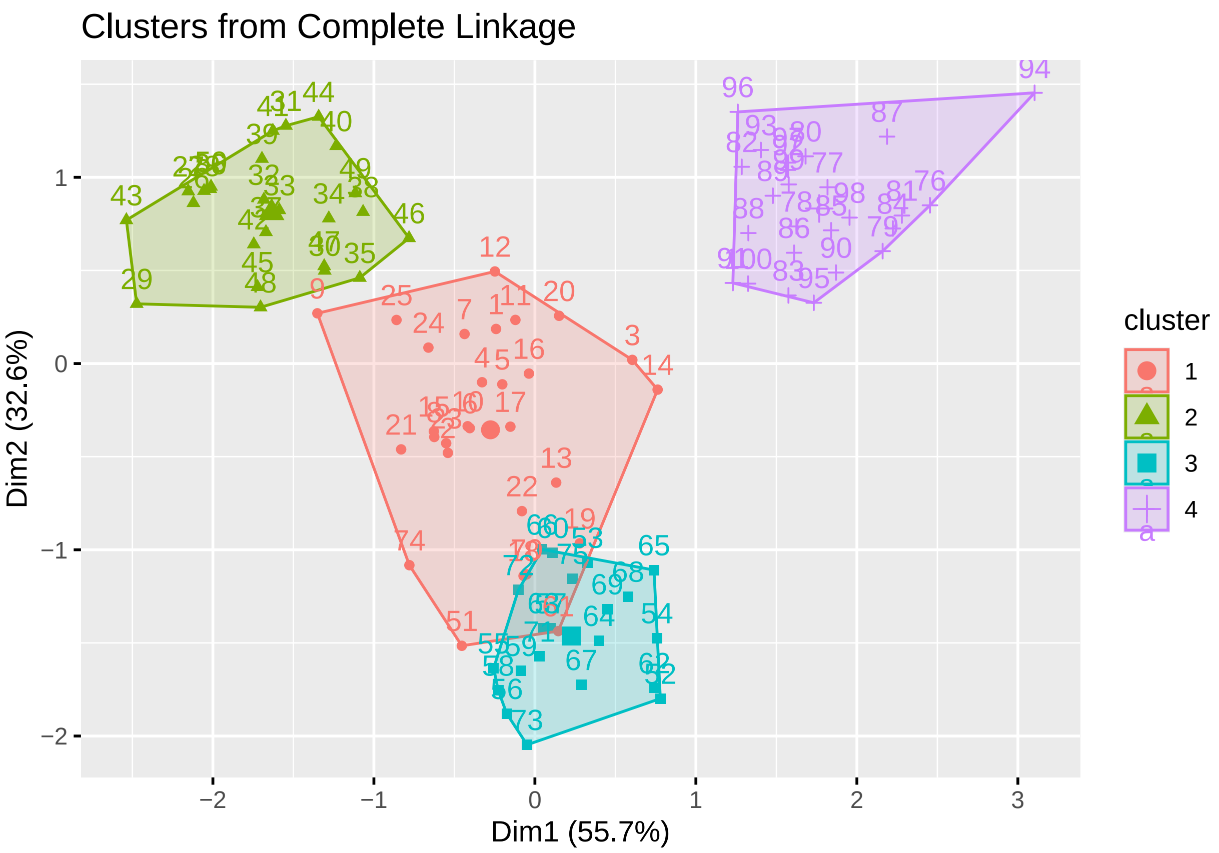

Silhouette Plot − WardD Linkage
n = 100
4 clusters C_j
j : n_j | $ave_{i \in Cj}$ s_i
1 : 24 | 0.45
2 : 26 | 0.51
3 : 25 | 0.60
4 : 25 | 0.62
Silhouette Width
0.0
0.2
0.4
0.6
0.8
1.0
Cluster
Average silhouette width : 0.55



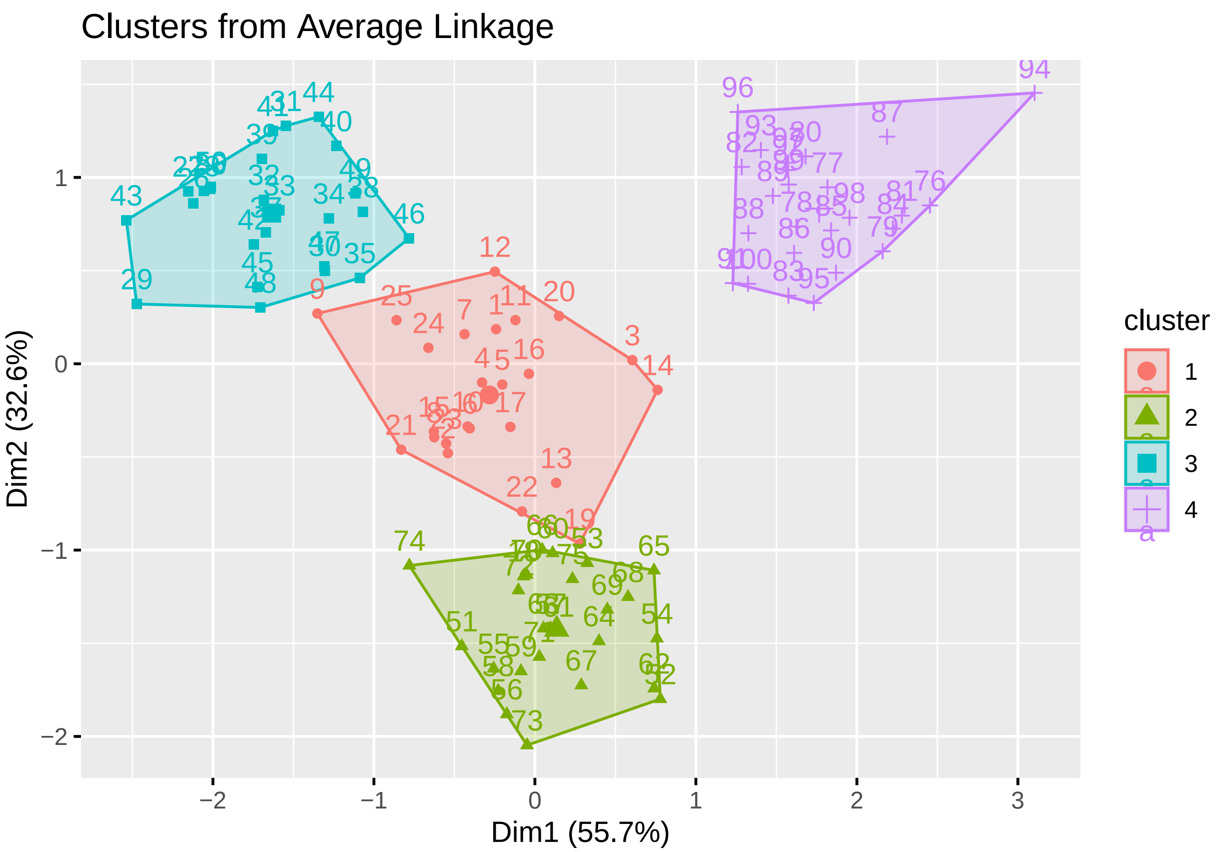
```
#Visualize Clusters from Different Linkage Methods
fviz_cluster(list(data = Xscaled, cluster = clusters_single),   main = "Clusters from Single Linkage")
```



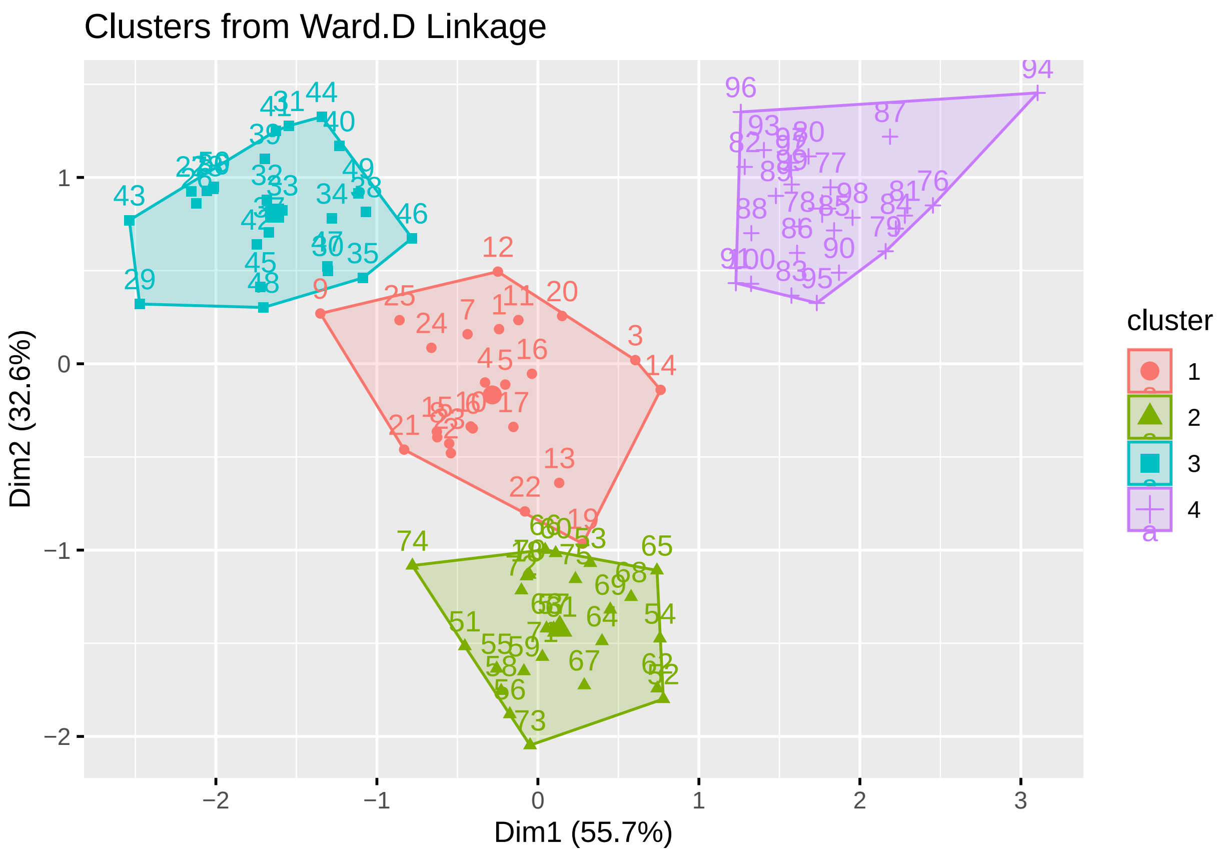
```
fviz_cluster(list(data = Xscaled, cluster = clusters_complete), main = "Clusters from Complete Linkage"
```



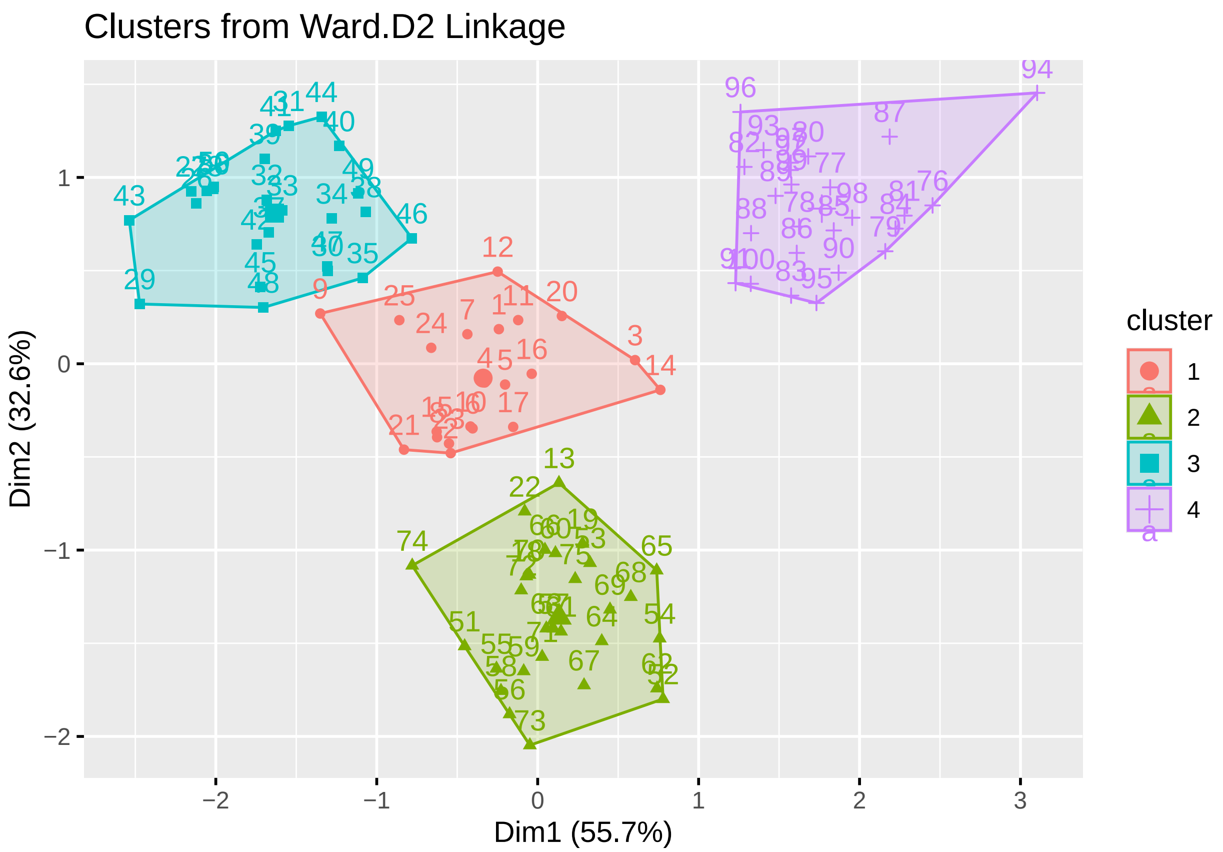
```
fviz_cluster(list(data = Xscaled, cluster = clusters_average),  main = "Clusters from Average Linkage")
```



```
fviz_cluster(list(data = Xscaled, cluster = clusters_wardD),    main = "Clusters from Ward.D Linkage")
```



```
fviz_cluster(list(data = Xscaled, cluster = clusters_wardD2),   main = "Clusters from Ward.D2 Linkage")
```



Internal Validation Silhouette and Dunn indices were calculated to assess cohesion vs. separation:

```
#internal validation Using silhouette and Dunn Index
#Silhouette for each method
library(cluster)

sil_single   <- mean(silhouette(clusters_single,   D)[, 3])
sil_complete <- mean(silhouette(clusters_complete, D)[, 3])
sil_average  <- mean(silhouette(clusters_average,  D)[, 3])
sil_wardD    <- mean(silhouette(clusters_wardD,    D)[, 3])
sil_wardD2   <- mean(silhouette(clusters_wardD2,   D)[, 3])

#Dunn Index for each method
library(clusterCrit)

# Convert scaled data to matrix
X_mat <- as.matrix(Xscaled)

# Compute Dunn index
dunn_single   <- intCriteria(X_mat, as.integer(clusters_single),   "Dunn")$dunn
dunn_complete <- intCriteria(X_mat, as.integer(clusters_complete), "Dunn")$dunn
dunn_average  <- intCriteria(X_mat, as.integer(clusters_average),  "Dunn")$dunn
dunn_wardD    <- intCriteria(X_mat, as.integer(clusters_wardD),    "Dunn")$dunn
dunn_wardD2   <- intCriteria(X_mat, as.integer(clusters_wardD2),   "Dunn")$dunn
```

External Validation

```
#External validation - using df$label
library(mclust)
```

```
## Package 'mclust' version 6.1.2
## Type 'citation("mclust")' for citing this R package in publications.
```

```
##
## Attaching package: 'mclust'
```

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

```
## The following object is masked from 'package:purrr':
##
##     map
```

```
#ARI (Adjusted Rand Index - Compares to True Labels)
ari_single   <- adjustedRandIndex(df$label, clusters_single)
ari_complete <- adjustedRandIndex(df$label, clusters_complete)
ari_average  <- adjustedRandIndex(df$label, clusters_average)
ari_wardD    <- adjustedRandIndex(df$label, clusters_wardD)
ari_wardD2   <- adjustedRandIndex(df$label, clusters_wardD2)
```

Summary of Results

```
#Summarize Results
results <- data.frame(
  Method     = c("Single", "Complete", "Average", "Ward.D", "Ward.D2"),
  Silhouette = round(c(sil_single, sil_complete, sil_average, sil_wardD, sil_wardD2), 3),
  Dunn_Index = round(c(dunn_single, dunn_complete, dunn_average, dunn_wardD, dunn_wardD2), 3),
  ARI        = round(c(ari_single, ari_complete, ari_average, ari_wardD, ari_wardD2), 3)
)

results
```

```
##     Method Silhouette Dunn_Index   ARI
## 1   Single      0.225      0.192 0.315
## 2 Complete      0.513      0.121 0.899
## 3  Average      0.545      0.191 0.973
## 4   Ward.D      0.545      0.191 0.973
## 5  Ward.D2      0.528      0.166 0.899
```

Cluster performance was evaluated using three criteria: Silhouette Index: Highest for Average and Ward.D methods (0.55), indicating well-separated clusters. Dunn Index: Consistent with silhouette trends; higher values show tighter and well-separated groups. ARI (Adjusted Rand Index): Ward.D and Average linkages achieved the strongest agreement (0.97) with the ground truth.

Results Interpretation: The Ward.D and Average linkage methods provided the most reliable and stable partitions, forming clusters similar to the original groups. Single linkage performed poorly due to chaining effects, while Complete linkage performed reasonably but slightly less efficiently than Ward.D2.

Visualization Summary • Dendrograms clearly showed separation between four clusters. • Gap statistic plot confirmed the “elbow” at k = 4. • Silhouette plots validated that clusters produced by Ward.D and Average were compact and well-defined. • Cluster scatterplots (from fviz_cluster) illustrated clear group boundaries for Ward methods.

Conclusion: Hierarchical clustering effectively identified the four natural groups in the dataset. Ward.D and Average linkage methods were superior, offering the highest internal cohesion and external alignment with true labels. The combination of silhouette analysis, Dunn index, and Adjusted Rand Index provided a comprehensive validation framework.