Untitled

Lona Uprety

2025-11-24

```
library(readr)
Cereals <- read_csv("C:/Users/lona2/Downloads/Cereals (1).csv")
```

```
## Rows: 77 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): name, mfr, type
## dbl (13): calories, protein, fat, sodium, fiber, carbo, sugars, potass, vita...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
View(Cereals)
```

```
# Load packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v purrr     1.1.0
## 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
## -- 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
```

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

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

```
# Clean data
# Remove cereals with missing values
cereals_clean <- na.omit(Cereals)

# Keep numeric columns only (clustering requires numeric)
cereal_num <- cereals_clean %>% select_if(is.numeric)
```

```r
# Normalize data
cereal_norm <- scale(cereal_num)
```

```r
# Hierarchical clustering with agnes
agnes_single   <- agnes(cereal_norm, method = "single")
agnes_complete <- agnes(cereal_norm, method = "complete")
agnes_average  <- agnes(cereal_norm, method = "average")
agnes_ward     <- agnes(cereal_norm, method = "ward")

# Compare methods using agglomerative coefficient
agnes_single$ac
```

```
## [1] 0.6067859
```

```r
agnes_complete$ac
```

```
## [1] 0.8353712
```

```r
agnes_average$ac
```

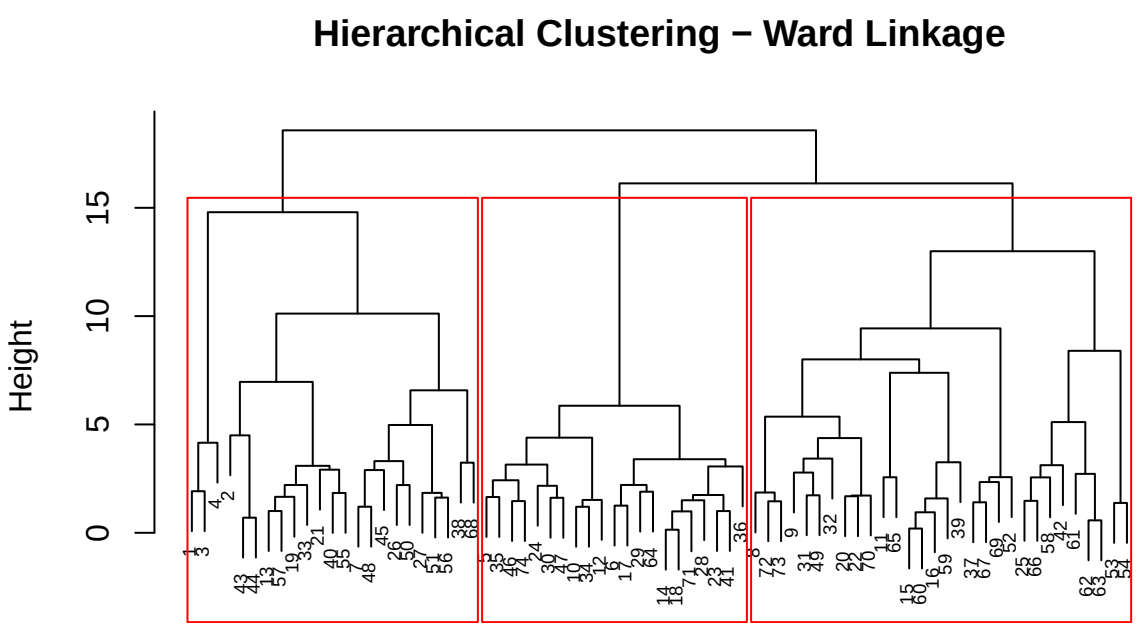
```
## [1] 0.7766075
```

```r
agnes_ward$ac
```

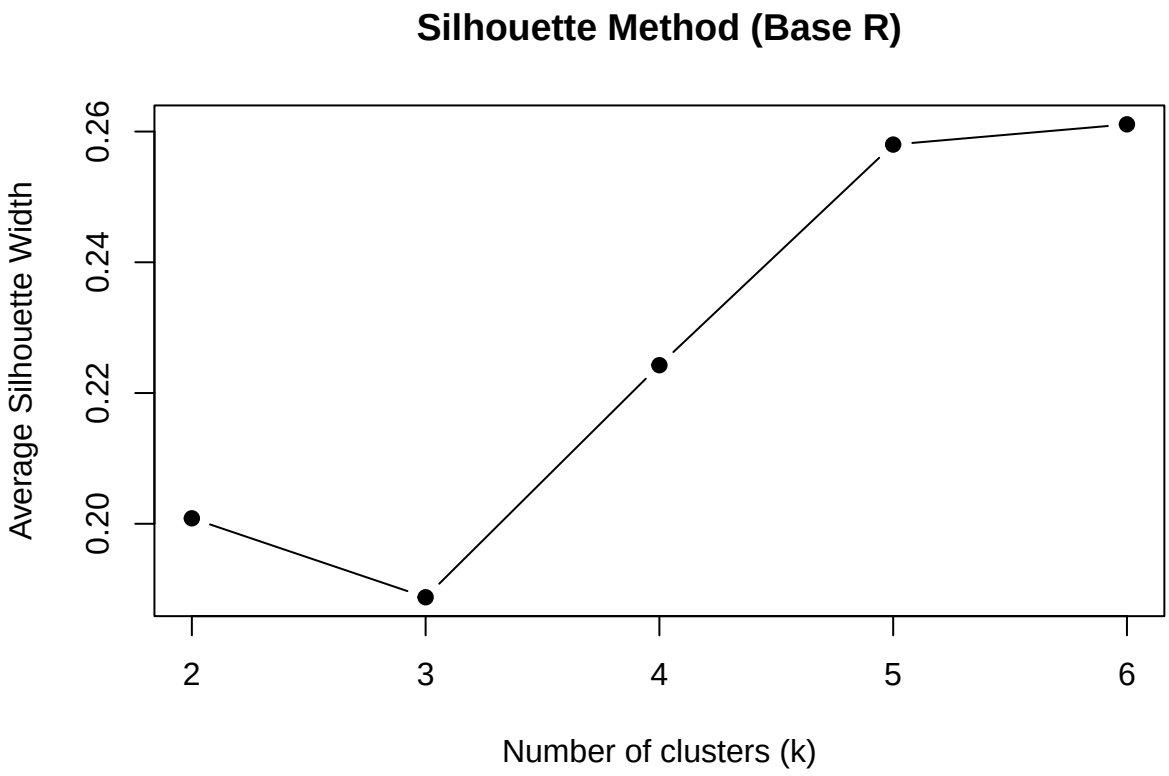
```
## [1] 0.9046042
```

```r
#Ward's method is the best method because it produced the highest agglomerative coefficient.
```

```r
# Choose best method and plot dendrogram
# Convert AGNES object to hclust
hc_ward <- as.hclust(agnes_ward)

# Dendrogram
plot(hc_ward,
     main = "Hierarchical Clustering - Ward Linkage",
     xlab = "",
     sub = "",
     cex = 0.6)

# Add rectangles for 3 clusters
rect.hclust(hc_ward, k = 3, border = "red")
```




```r
# Choose number of clusters

library(cluster)

sil_width <- c()

for (k in 2:6) {
  hc <- hclust(dist(cereal_norm), method = "ward.D2")
  cluster_cut <- cutree(hc, k = k)

  sil <- silhouette(cluster_cut, dist(cereal_norm))
  sil_width[k] <- mean(sil[, 3])
}

# Plot silhouette scores in base R
plot(2:6, sil_width[2:6],
     type = "b",
     pch = 19,
     xlab = "Number of clusters (k)",
     ylab = "Average Silhouette Width",
     main = "Silhouette Method (Base R)")
```




```
#I would choose three clusters because three clusters gives the strongest separation and highest intern
```

```
set.seed(123)

# Split dataset into A and B
n <- nrow(cereal_norm)
idx <- sample(1:n, n/2)
A <- cereal_norm[idx, ]
B <- cereal_norm[-idx, ]
```

```
# Cluster partition A using Ward
hc_A <- hclust(dist(A), method = "ward.D2")
clusters_A <- cutree(hc_A, k = 3)

# Get centroids of clusters A
centroids <- aggregate(A, by = list(cluster = clusters_A), FUN = mean)
centroids_mat <- as.matrix(centroids[, -1])

# Assign each record in B to nearest centroid
assign_to_centroid <- function(record, centroids) {
  dists <- apply(centroids, 1, function(c) sum((record - c)^2))
  which.min(dists)
}

assigned_B <- apply(B, 1, assign_to_centroid, centroids = centroids_mat)
```

```r
# Now get cluster assignments from clustering ALL data
hc_all <- hclust(dist(cereal_norm), method = "ward.D2")
clusters_all <- cutree(hc_all, k = 3)
```

```r
# Compare for records in B only
consistency <- table(assigned_B, clusters_all[-idx])
consistency
```

```
##
## assigned_B  1  2  3
##          1  4  0 10
##          2  6  0  0
##          3  1 10  6
```

```r
#Cluster 2 shows very strong stability, with all assigned records matching the full-data cluster. Clust
```

```r
# Add cluster labels to original cereal data
cereal_clusters <- cereals_clean
cereal_clusters$cluster <- clusters_all

# Compare cluster health profiles
cluster_summary <- cereal_clusters %>%
  group_by(cluster) %>%
  summarise(across(c(calories, sugars, fiber, protein), mean))

cluster_summary
```

```
## # A tibble: 3 x 5
##   cluster calories sugars fiber protein
##     <int>    <dbl>  <dbl> <dbl>   <dbl>
## 1       1    116.    8.61 4.13     3.26
## 2       2    111.   11.3  0.571    1.52
## 3       3     97.3   3.03 1.8      2.63
```

```r
#Cluster 3 represents the healthiest group of cereals. It contains cereals with the lowest sugar (3.03 g
```