This is an [R Markdown](http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Cmd+Shift+Enter*.

#------Load Librarys,data and prepocess to omit missing values  
library("tidyverse")

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.2 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.4 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.4   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library("stats")  
library("cluster")  
library ("factoextra")

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

cereals<-read.csv("/Users/hollyvictor/Desktop/Cereals.csv")  
cereals<-na.omit(cereals)  
  
#----- Normalize the Data   
numeric\_cereals<-cereals %>% select(where(is.numeric))  
scaled\_cereals<-scale(numeric\_cereals)  
  
#------Apply Hierarchial clustering using Agnes  
agnes\_single <- agnes(scaled\_cereals, method = "single")  
agnes\_complete <- agnes(scaled\_cereals, method = "complete")  
agnes\_average <- agnes(scaled\_cereals, method = "average")  
agnes\_ward <- agnes(scaled\_cereals, method = "ward")  
  
agnes\_single$ac

## [1] 0.6067859

agnes\_complete$ac

## [1] 0.8353712

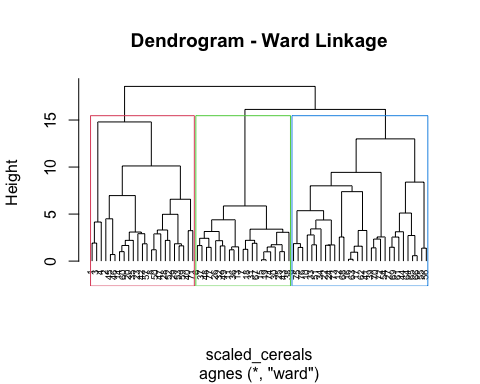
agnes\_average$ac

## [1] 0.7766075

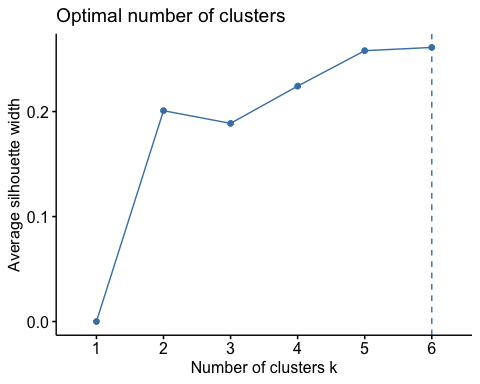
agnes\_ward$ac

## [1] 0.9046042

#-----Visualize Dendogram and Choose Number of Clusters  
pltree(agnes\_ward, cex = 0.6, hang = -1, main = "Dendrogram - Ward Linkage")  
rect.hclust(as.hclust(agnes\_ward), k = 3, border = 2:5)

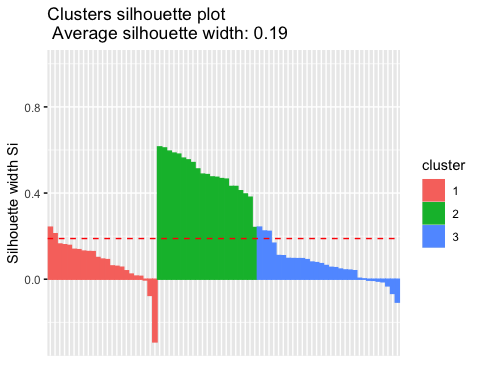


#Silhouette Analysis to Justify k = 3  
library(factoextra)  
  
# Silhouette method to help choose optimal k  
fviz\_nbclust(scaled\_cereals, FUN = hcut, method = "silhouette", k.max = 6)



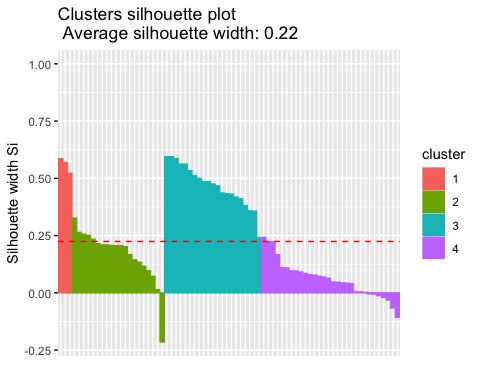
# Visualize silhouette for k = 3  
hclust\_ward <- hcut(scaled\_cereals, k = 3, hc\_method = "ward.D2")  
fviz\_silhouette(hclust\_ward)

## cluster size ave.sil.width  
## 1 1 23 0.08  
## 2 2 21 0.49  
## 3 3 30 0.06



# Visualize silhouette for k = 4  
hclust\_ward\_k4 <- hcut(scaled\_cereals, k = 4, hc\_method = "ward.D2")  
fviz\_silhouette(hclust\_ward\_k4)

## cluster size ave.sil.width  
## 1 1 3 0.56  
## 2 2 20 0.17  
## 3 3 21 0.47  
## 4 4 30 0.06



#-----Evaluate Cluster Stability  
set.seed(42)  
rows <- nrow(scaled\_cereals)  
idx <- sample(1:rows, size = floor(rows/2))  
partA <- scaled\_cereals[idx, ]  
partB <- scaled\_cereals[-idx, ]  
  
#----Cluster partition A  
agnes\_A <- agnes(partA, method = "ward")  
clust\_A <- cutree(as.hclust(agnes\_A), k = 3)  
  
#----Calculate Centroids from A  
centroids<-aggregate(partA,by=list(cluster=clust\_A),mean)[-1]  
  
#Assign Part B to nearest centroid  
assign\_cluster <- function(obs, centers) {  
 apply(centers, 1, function(center) sum((obs - center)^2)) %>% which.min()  
}  
assigned\_B <- apply(partB, 1, assign\_cluster, centers = centroids)  
  
full\_clusters <- cutree(as.hclust(agnes\_ward), k = 3)  
  
#-----Optional Consistency Check  
mean(assigned\_B == full\_clusters[-idx])

## [1] 0.2972973

#-----Optional Consistency Check for k = 4  
# Evaluate cluster stability for k = 4  
agnes\_A\_k4 <- agnes(partA, method = "ward")  
clust\_A\_k4 <- cutree(as.hclust(agnes\_A\_k4), k = 4)  
  
# Calculate centroids for k = 4  
centroids\_k4 <- aggregate(partA, by = list(cluster = clust\_A\_k4), mean)[-1]  
  
# Assign Part B to nearest centroid using k = 4  
assigned\_B\_k4 <- apply(partB, 1, assign\_cluster, centers = centroids\_k4)  
  
# Compare with full clustering using k = 4  
full\_clusters\_k4 <- cutree(as.hclust(agnes\_ward), k = 4)  
  
# Stability score for k = 4  
mean(assigned\_B\_k4 == full\_clusters\_k4[-idx])

## [1] 0.02702703

#-----Identify the Healthy Cereal Cluster  
cereal\_clusters <- cutree(as.hclust(agnes\_ward), k = 3)  
cereals$Cluster <- cereal\_clusters  
  
aggregate(cereals[, c("sugars", "fiber", "calories", "rating")],  
 by = list(Cluster = cereals$Cluster), FUN = mean)

## Cluster sugars fiber calories rating  
## 1 1 8.608696 4.1304348 116.08696 42.90285  
## 2 2 11.285714 0.5714286 110.95238 28.84825  
## 3 3 3.033333 1.8000000 97.33333 51.43111

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Cmd+Option+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Cmd+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.