Ayrianna Teachout

2023-12-04

# Part 1 - Clustering

## K-means Clustering

### Part a - Best value of k using an elbow plot.

library(stats)  
library(ggplot2)  
library(factoextra)

## Warning: package 'factoextra' was built under R version 4.3.2

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

library(corrplot)

## Warning: package 'corrplot' was built under R version 4.3.2

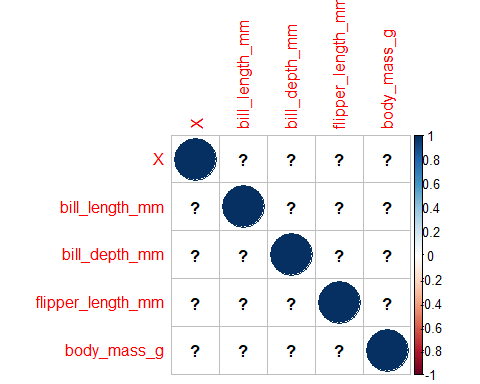
## corrplot 0.92 loaded

library(cluster)  
avianmeasurements\_data <- read.csv("AvianMeasurements.csv")

#### Check for correlation among variables.

The graph below shows there is no correlation between variables.

corrplot(cor(avianmeasurements\_data))

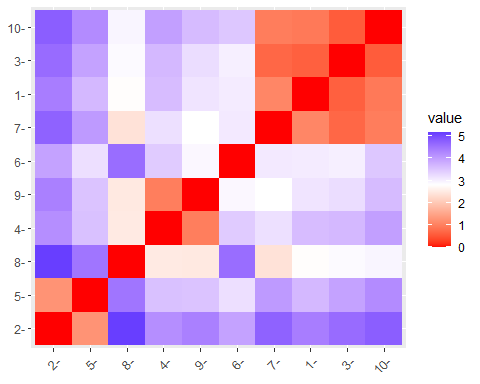


#### Elbow Plot

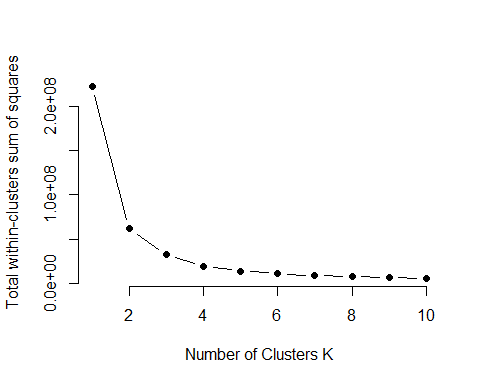
df <- scale(avianmeasurements\_data)  
set.seed(123)  
  
# Sample 10 rows from the scaled data  
avia <- sample(1:nrow(df), 10)  
avia1 <- df[avia,]  
  
distE <- dist(avia1, method='euclidean')  
print(round(as.matrix(distE)[1:4, 1:4], 1))

## 1 2 3 4  
## 1 0.0 4.4 0.6 3.6  
## 2 4.4 0.0 4.6 4.2  
## 3 0.6 4.6 0.0 3.7  
## 4 3.6 4.2 3.7 0.0

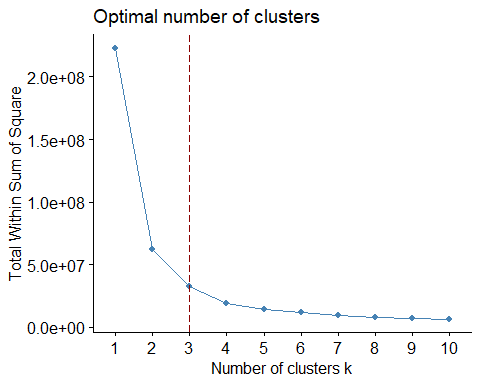
# Visualize the distance matrix  
fviz\_dist(distE)



# Omit missing values  
avianmeasurements\_data <- na.omit(avianmeasurements\_data)  
  
k\_values <- 1:10  
  
  
# Calculate the total within-cluster sum of squares for each k  
wss <- sapply(k\_values, function(k) {  
 kmeans\_model <- kmeans(avianmeasurements\_data, centers = k, nstart = 20, iter.max = 15)  
 return(kmeans\_model$tot.withinss)  
})  
  
# Plot the elbow plot  
plot(k\_values, wss, type="b", pch=19, frame=FALSE, xlab="Number of Clusters K", ylab="Total within-clusters sum of squares")



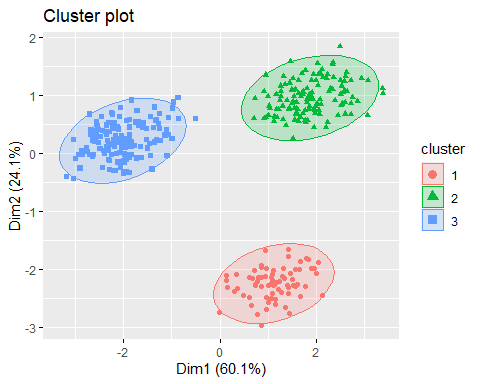
fviz\_nbclust(avianmeasurements\_data, kmeans, method="wss") + geom\_vline(xintercept=3, linetype=5, col="darkred")



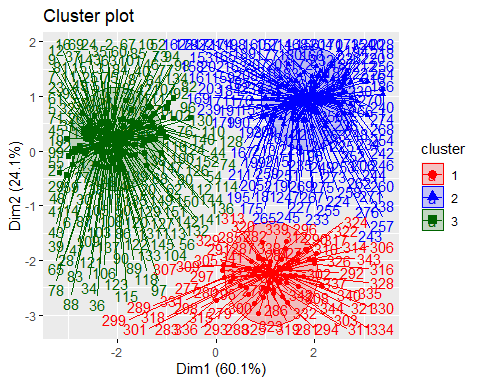
Optimal k-value would be 3 determined by the elbow plot. In the elbow plot, you can observe a rapid decrease in the within-cluster sum of squares as it increased the number of clusters from 1 to 3. However, beyond k=3, the rate of decrease started to slow down, forming an ‘elbow’ in the plot. This suggests that k=3 is a reasonable choice, as it captures a significant portion of the variability in the data while avoiding excessive complexity.

### Part b - Cluster Visualization

avianmeasurements\_data <- na.omit(avianmeasurements\_data)  
  
df <- na.omit(df)  
km.res <- kmeans(df, 3, nstart = 25)  
  
nrow\_cluster <- length(km.res$cluster)  
  
avianmeasurements\_data <- avianmeasurements\_data[1:nrow\_cluster, ]  
  
df\_member <- cbind(avianmeasurements\_data, cluster = km.res$cluster)  
  
# Visualize clusters without deprecated arguments  
fviz\_cluster(km.res, data = df\_member, geom = "point", ellipse.type = "norm")



fviz\_cluster(km.res, data = df\_member,  
 palette=c("red", "blue", "darkgreen"),  
 ellipse.type = "euclid",  
 star.plot = T,  
 repel = T,  
 ggtheme = theme())



### Part c - Mean, median and mode of clusters

avianmeasurements\_data <- na.omit(avianmeasurements\_data)  
km.res <- kmeans(avianmeasurements\_data, centers = 3, nstart = 20, iter.max = 15)  
df\_member <- cbind(avianmeasurements\_data, cluster = km.res$cluster)  
  
#Mean  
mean\_results <- aggregate(. ~ cluster, data = as.data.frame(df\_member), mean)  
print(mean\_results)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 213.9706 49.05000 15.56618 220.7647 5449.632  
## 2 2 149.4615 41.19882 18.04734 190.2189 3529.586  
## 3 3 183.3810 44.98381 16.73524 205.2762 4475.476

#Median   
median\_results <- aggregate(. ~ cluster, data = as.data.frame(df\_member), median)  
print(median\_results)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 213 49.15 15.65 220.5 5425  
## 2 2 117 39.60 18.00 190.0 3550  
## 3 3 196 45.30 15.70 208.0 4450

#Mode   
mode\_results <- aggregate(. ~ cluster, data = as.data.frame(df\_member), mode)  
  
print(mode\_results)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 numeric numeric numeric numeric numeric  
## 2 2 numeric numeric numeric numeric numeric  
## 3 3 numeric numeric numeric numeric numeric

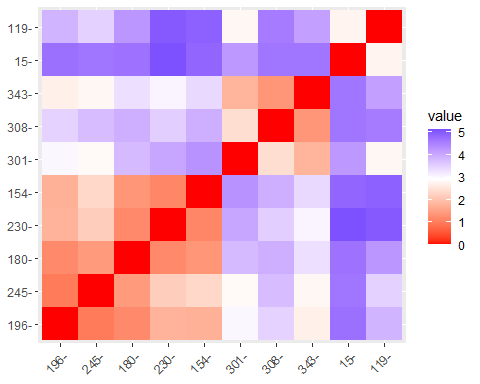
### Part d - Another k-value Analysis

##### Elbow Plot 2

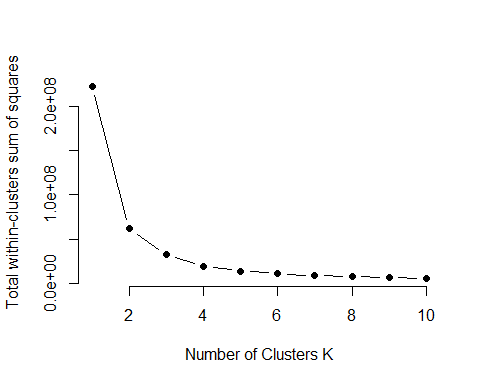
df <- scale(avianmeasurements\_data)  
set.seed(123)  
  
# Sample 10 rows from the scaled data  
avia <- sample(1:nrow(df), 10)  
avia1 <- df[avia,]  
  
distE <- dist(avia1, method='euclidean')  
print(round(as.matrix(distE)[1:4, 1:4], 1))

## 180 15 196 308  
## 180 0.0 4.7 1.2 3.9  
## 15 4.7 0.0 4.7 4.7  
## 196 1.2 4.7 0.0 3.5  
## 308 3.9 4.7 3.5 0.0

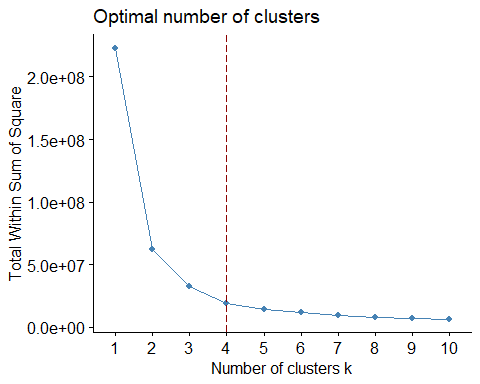
# Visualize the distance matrix  
fviz\_dist(distE)



# Omit missing values  
avianmeasurements\_data <- na.omit(avianmeasurements\_data)  
  
k\_values <- 1:10  
  
  
# Calculate the total within-cluster sum of squares for each k  
wss <- sapply(k\_values, function(k) {  
 kmeans\_model <- kmeans(avianmeasurements\_data, centers = k, nstart = 20, iter.max = 15)  
 return(kmeans\_model$tot.withinss)  
})  
  
# Plot the elbow plot  
plot(k\_values, wss, type="b", pch=19, frame=FALSE, xlab="Number of Clusters K", ylab="Total within-clusters sum of squares")



fviz\_nbclust(avianmeasurements\_data, kmeans, method="wss") + geom\_vline(xintercept=4, linetype=5, col="darkred")



Optimal k-value would be 4 determined by the elbow plot. In the elbow plot, you can observe a rapid decrease in the within-cluster sum of squares as it increased the number of clusters from 1 to 4. However, beyond k=4, the rate of decrease started to slow down, forming an ‘elbow’ in the plot. This suggests that k=4 is a reasonable choice, as it captures a significant portion of the variability in the data while avoiding excessive complexity.

#### Cluster Visualization

df[is.na(df)] <- 0  
df[is.infinite(df)] <- NA  
km.res <- kmeans(df, 4, nstart = 25)  
km.res

## K-means clustering with 4 clusters of sizes 151, 66, 57, 68  
##   
## Cluster means:  
## X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 -0.9651795 -0.9397308 0.6052219 -0.7795325 -0.6248386  
## 2 0.3904463 0.2943188 -1.4292037 0.8669538 0.6123148  
## 3 0.4471984 1.0753664 -0.7153018 1.4932089 1.6434628  
## 4 1.3894461 0.8996828 0.6428109 -0.3620949 -0.5844049  
##   
## Clustering vector:  
## 1 2 3 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161   
## 1 1 1 1 1 1 1 1 1 1 1 2 3 2 3 3 2 2 3 2   
## 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181   
## 2 2 3 2 3 2 3 2 3 2 3 3 2 2 2 2 2 2 3 2   
## 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201   
## 3 3 2 2 3 3 3 2 3 2 3 2 3 2 2 3 2 2 3 2   
## 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221   
## 3 2 3 2 3 2 2 2 2 2 3 2 3 2 3 2 3 2 3 2   
## 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241   
## 3 2 3 3 2 2 3 2 3 2 3 2 3 2 3 2 3 2 3 2   
## 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261   
## 3 2 3 2 3 2 3 3 2 2 3 2 3 2 3 2 3 2 3 2   
## 262 263 264 265 266 267 268 269 270 271 273 274 275 276 277 278 279 280 281 282   
## 3 3 3 2 3 2 3 2 3 2 2 3 2 3 4 4 4 4 4 4   
## 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302   
## 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4   
## 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322   
## 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4   
## 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342   
## 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4   
## 343 344   
## 4 4   
##   
## Within cluster sum of squares by cluster:  
## [1] 203.33439 35.32869 44.77511 82.49863  
## (between\_SS / total\_SS = 78.5 %)  
##   
## Available components:  
##   
## [1] "cluster" "centers" "totss" "withinss" "tot.withinss"  
## [6] "betweenss" "size" "iter" "ifault"

km.res$totss

## [1] 1705

km.res$betweenss

## [1] 1339.063

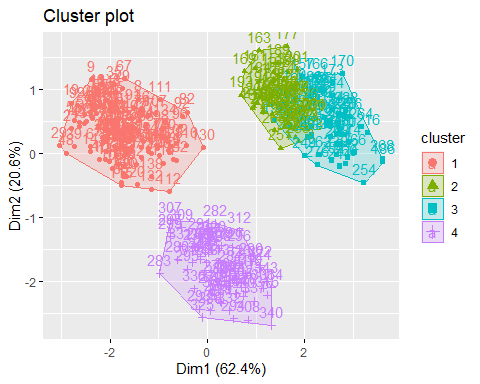
km.res$betweenss/km.res$totss

## [1] 0.7853743

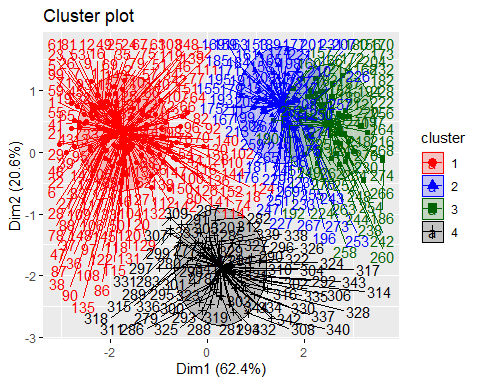
df\_member <- cbind(avianmeasurements\_data, cluster = km.res$cluster)  
head(df\_member)

## X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g cluster  
## 1 1 39.1 18.7 181 3750 1  
## 2 2 39.5 17.4 186 3800 1  
## 3 3 40.3 18.0 195 3250 1  
## 5 5 36.7 19.3 193 3450 1  
## 6 6 39.3 20.6 190 3650 1  
## 7 7 38.9 17.8 181 3625 1

fviz\_cluster(km.res, data=avianmeasurements\_data)



fviz\_cluster(km.res, data = avianmeasurements\_data,  
 palette=c("red", "blue", "darkgreen", "black"),  
 ellipse.type = "euclid",  
 star.plot = T,  
 repel = T,  
 ggtheme = theme())



aggregate(avianmeasurements\_data, by=list(cluster=df\_member$cluster), mean)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 76.98013 38.79139 18.34636 189.9536 3700.662  
## 2 2 211.42424 45.52879 14.32879 213.1061 4692.803  
## 3 3 217.05263 49.79298 15.73860 221.9123 5519.737  
## 4 4 310.50000 48.83382 18.42059 195.8235 3733.088

aggregate(avianmeasurements\_data, by=list(cluster=df\_member$cluster), median)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 77.0 38.80 18.40 190 3700.0  
## 2 2 209.5 45.50 14.35 213 4712.5  
## 3 3 220.0 49.80 15.80 222 5500.0  
## 4 4 310.5 49.55 18.45 196 3700.0

aggregate(avianmeasurements\_data, by=list(cluster=df\_member$cluster), mode)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 numeric numeric numeric numeric numeric  
## 2 2 numeric numeric numeric numeric numeric  
## 3 3 numeric numeric numeric numeric numeric  
## 4 4 numeric numeric numeric numeric numeric

#### Mean, median and mode of clusters

#Row 272 and row 3 contain missing values.  
avianmeasurements\_data <- na.omit(avianmeasurements\_data)  
km.res <- kmeans(avianmeasurements\_data, centers = 4, nstart = 20, iter.max = 15)  
df\_member <- cbind(avianmeasurements\_data, cluster = km.res$cluster)  
  
#Mean  
mean\_results <- aggregate(. ~ cluster, data = as.data.frame(df\_member), mean)  
print(mean\_results)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 194.0375 45.34250 15.69125 209.7125 4708.125  
## 2 2 215.7818 49.67455 15.72545 221.7091 5548.182  
## 3 3 139.5918 39.98980 17.75204 188.4796 3325.510  
## 4 4 165.0734 43.51193 18.40183 195.1468 3938.532

#Median   
median\_results <- aggregate(. ~ cluster, data = as.data.frame(df\_member), median)  
print(median\_results)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 207.5 45.55 14.6 212.0 4700  
## 2 2 218.0 49.60 15.8 222.0 5550  
## 3 3 107.5 38.75 17.8 188.5 3350  
## 4 4 130.0 42.00 18.7 194.0 3900

#Mode   
mode\_results <- aggregate(. ~ cluster, data = as.data.frame(df\_member), mode)  
  
print(mode\_results)

## cluster X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 numeric numeric numeric numeric numeric  
## 2 2 numeric numeric numeric numeric numeric  
## 3 3 numeric numeric numeric numeric numeric  
## 4 4 numeric numeric numeric numeric numeric

### Part e - Comparison and Justification

The data above shows that the optimal k-value is 3. Looking at both elbow plots, you can conclude that 3 clusters is beneficial because the rate of decrease in WSS slows down significantly more than at the k-value of 4. You can also see the overlap in clusters in analysis #2 in the cluster visualization. This proves that the k-value of 4 is too large and creates more clusters than actually exist in the data.

## Hierarchical clustering (AGNES)

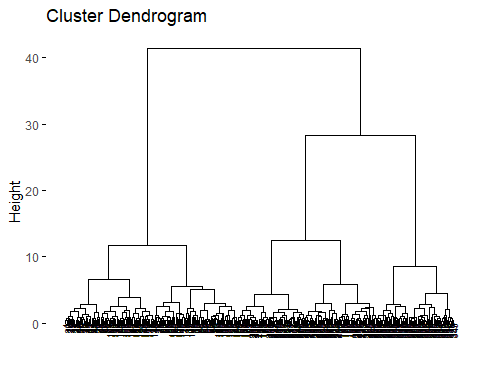
### Part a - Dendrogram

library(factoextra)  
library(cluster)  
  
avianmeasurements\_scaled <- scale(avianmeasurements\_data)  
  
#### Find the dis(similarity)   
res.dist <- dist(avianmeasurements\_scaled, method = "euclidean")  
as.matrix(res.dist)[1:6, 1:6] #Distances between first 6 observations

## 1 2 3 5 6 7  
## 1 0.0000000 0.7544172 1.2467275 1.0748867 1.1639438 0.4868263  
## 2 0.7544172 0.0000000 0.9969384 1.2758963 1.6566960 0.4792699  
## 3 1.2467275 0.9969384 0.0000000 0.9751835 1.4639329 1.1346975  
## 5 1.0748867 1.2758963 0.9751835 0.0000000 0.8763334 1.2311026  
## 6 1.1639438 1.6566960 1.4639329 0.8763334 0.0000000 1.5577042  
## 7 0.4868263 0.4792699 1.1346975 1.2311026 1.5577042 0.0000000

### Conduct the HC with the agnes()  
res.agnes <- agnes(avianmeasurements\_scaled, method = "ward")  
  
#### Visualize the dendrogram  
fviz\_dend(res.agnes, cex = 0.5)

## 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.



### Part b - Cophentic Correlation

### Finding the cophenetic distances (height)  
res.coph <- cophenetic(res.agnes)  
  
### Comparing cophenetic distances with original distances  
cor(res.dist, res.coph)

## [1] 0.8391831

The cophenetic correlation is 0.8391831, suggesting that the hierarchical clustering dendrogram provides a strong representation of the original distances in the AvianMeasurements dataset.

### Part c/d - K-value Dendrogram

The dendrogram above shows three distinct clusters on the dendrogram. Showing the appropriate value of k would be 3.

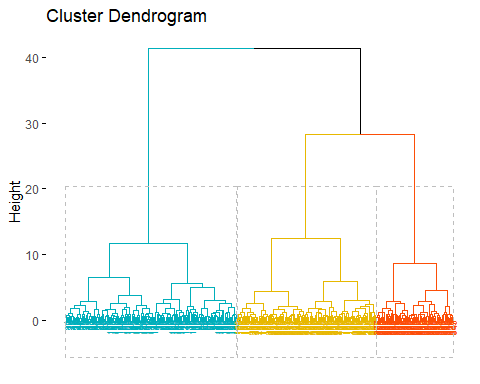
#### Cutting the dendrogram  
k <- 3 # You can choose an appropriate value based on the dendrogram  
grp <- cutree(res.agnes, k = k)  
  
#### Identifying groups  
head(grp)

## [1] 1 1 1 1 1 1

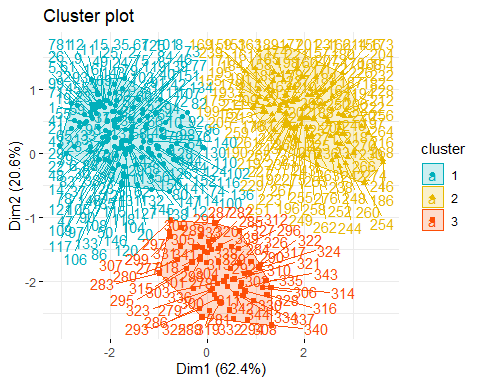
# Displaying some observations from the first group  
avianmeasurements\_data[grp == 1, ]

## X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 39.1 18.7 181 3750  
## 2 2 39.5 17.4 186 3800  
## 3 3 40.3 18.0 195 3250  
## 5 5 36.7 19.3 193 3450  
## 6 6 39.3 20.6 190 3650  
## 7 7 38.9 17.8 181 3625  
## 8 8 39.2 19.6 195 4675  
## 9 9 34.1 18.1 193 3475  
## 10 10 42.0 20.2 190 4250  
## 11 11 37.8 17.1 186 3300  
## 12 12 37.8 17.3 180 3700  
## 13 13 41.1 17.6 182 3200  
## 14 14 38.6 21.2 191 3800  
## 15 15 34.6 21.1 198 4400  
## 16 16 36.6 17.8 185 3700  
## 17 17 38.7 19.0 195 3450  
## 18 18 42.5 20.7 197 4500  
## 19 19 34.4 18.4 184 3325  
## 20 20 46.0 21.5 194 4200  
## 21 21 37.8 18.3 174 3400  
## 22 22 37.7 18.7 180 3600  
## 23 23 35.9 19.2 189 3800  
## 24 24 38.2 18.1 185 3950  
## 25 25 38.8 17.2 180 3800  
## 26 26 35.3 18.9 187 3800  
## 27 27 40.6 18.6 183 3550  
## 28 28 40.5 17.9 187 3200  
## 29 29 37.9 18.6 172 3150  
## 30 30 40.5 18.9 180 3950  
## 31 31 39.5 16.7 178 3250  
## 32 32 37.2 18.1 178 3900  
## 33 33 39.5 17.8 188 3300  
## 34 34 40.9 18.9 184 3900  
## 35 35 36.4 17.0 195 3325  
## 36 36 39.2 21.1 196 4150  
## 37 37 38.8 20.0 190 3950  
## 38 38 42.2 18.5 180 3550  
## 39 39 37.6 19.3 181 3300  
## 40 40 39.8 19.1 184 4650  
## 41 41 36.5 18.0 182 3150  
## 42 42 40.8 18.4 195 3900  
## 43 43 36.0 18.5 186 3100  
## 44 44 44.1 19.7 196 4400  
## 45 45 37.0 16.9 185 3000  
## 46 46 39.6 18.8 190 4600  
## 47 47 41.1 19.0 182 3425  
## 48 48 37.5 18.9 179 2975  
## 49 49 36.0 17.9 190 3450  
## 50 50 42.3 21.2 191 4150  
## 51 51 39.6 17.7 186 3500  
## 52 52 40.1 18.9 188 4300  
## 53 53 35.0 17.9 190 3450  
## 54 54 42.0 19.5 200 4050  
## 55 55 34.5 18.1 187 2900  
## 56 56 41.4 18.6 191 3700  
## 57 57 39.0 17.5 186 3550  
## 58 58 40.6 18.8 193 3800  
## 59 59 36.5 16.6 181 2850  
## 60 60 37.6 19.1 194 3750  
## 61 61 35.7 16.9 185 3150  
## 62 62 41.3 21.1 195 4400  
## 63 63 37.6 17.0 185 3600  
## 64 64 41.1 18.2 192 4050  
## 65 65 36.4 17.1 184 2850  
## 66 66 41.6 18.0 192 3950  
## 67 67 35.5 16.2 195 3350  
## 68 68 41.1 19.1 188 4100  
## 69 69 35.9 16.6 190 3050  
## 70 70 41.8 19.4 198 4450  
## 71 71 33.5 19.0 190 3600  
## 72 72 39.7 18.4 190 3900  
## 73 73 39.6 17.2 196 3550  
## 74 74 45.8 18.9 197 4150  
## 75 75 35.5 17.5 190 3700  
## 76 76 42.8 18.5 195 4250  
## 77 77 40.9 16.8 191 3700  
## 78 78 37.2 19.4 184 3900  
## 79 79 36.2 16.1 187 3550  
## 80 80 42.1 19.1 195 4000  
## 81 81 34.6 17.2 189 3200  
## 82 82 42.9 17.6 196 4700  
## 83 83 36.7 18.8 187 3800  
## 84 84 35.1 19.4 193 4200  
## 85 85 37.3 17.8 191 3350  
## 86 86 41.3 20.3 194 3550  
## 87 87 36.3 19.5 190 3800  
## 88 88 36.9 18.6 189 3500  
## 89 89 38.3 19.2 189 3950  
## 90 90 38.9 18.8 190 3600  
## 91 91 35.7 18.0 202 3550  
## 92 92 41.1 18.1 205 4300  
## 93 93 34.0 17.1 185 3400  
## 94 94 39.6 18.1 186 4450  
## 95 95 36.2 17.3 187 3300  
## 96 96 40.8 18.9 208 4300  
## 97 97 38.1 18.6 190 3700  
## 98 98 40.3 18.5 196 4350  
## 99 99 33.1 16.1 178 2900  
## 100 100 43.2 18.5 192 4100  
## 101 101 35.0 17.9 192 3725  
## 102 102 41.0 20.0 203 4725  
## 103 103 37.7 16.0 183 3075  
## 104 104 37.8 20.0 190 4250  
## 105 105 37.9 18.6 193 2925  
## 106 106 39.7 18.9 184 3550  
## 107 107 38.6 17.2 199 3750  
## 108 108 38.2 20.0 190 3900  
## 109 109 38.1 17.0 181 3175  
## 110 110 43.2 19.0 197 4775  
## 111 111 38.1 16.5 198 3825  
## 112 112 45.6 20.3 191 4600  
## 113 113 39.7 17.7 193 3200  
## 114 114 42.2 19.5 197 4275  
## 115 115 39.6 20.7 191 3900  
## 116 116 42.7 18.3 196 4075  
## 117 117 38.6 17.0 188 2900  
## 118 118 37.3 20.5 199 3775  
## 119 119 35.7 17.0 189 3350  
## 120 120 41.1 18.6 189 3325  
## 121 121 36.2 17.2 187 3150  
## 122 122 37.7 19.8 198 3500  
## 123 123 40.2 17.0 176 3450  
## 124 124 41.4 18.5 202 3875  
## 125 125 35.2 15.9 186 3050  
## 126 126 40.6 19.0 199 4000  
## 127 127 38.8 17.6 191 3275  
## 128 128 41.5 18.3 195 4300  
## 129 129 39.0 17.1 191 3050  
## 130 130 44.1 18.0 210 4000  
## 131 131 38.5 17.9 190 3325  
## 132 132 43.1 19.2 197 3500  
## 133 133 36.8 18.5 193 3500  
## 134 134 37.5 18.5 199 4475  
## 135 135 38.1 17.6 187 3425  
## 136 136 41.1 17.5 190 3900  
## 137 137 35.6 17.5 191 3175  
## 138 138 40.2 20.1 200 3975  
## 139 139 37.0 16.5 185 3400  
## 140 140 39.7 17.9 193 4250  
## 141 141 40.2 17.1 193 3400  
## 142 142 40.6 17.2 187 3475  
## 143 143 32.1 15.5 188 3050  
## 144 144 40.7 17.0 190 3725  
## 145 145 37.3 16.8 192 3000  
## 146 146 39.0 18.7 185 3650  
## 147 147 39.2 18.6 190 4250  
## 148 148 36.6 18.4 184 3475  
## 149 149 36.0 17.8 195 3450  
## 150 150 37.8 18.1 193 3750  
## 151 151 36.0 17.1 187 3700  
## 152 152 41.5 18.5 201 4000

#### Plotting using color  
fviz\_dend(res.agnes, k = k, cex = 0.5,   
 k\_colors = c("#00AFBB", "#E7B800", "#FC4E07"),   
 color\_labels\_by\_k = TRUE, rect = TRUE)



### Visualizing results as a scatter plot  
fviz\_cluster(list(data = avianmeasurements\_scaled, cluster = grp),   
 palette = c("#00AFBB", "#E7B800", "#FC4E07"),  
 ellipse.type = "convex", repel = TRUE, show.clust.cent = FALSE,   
 ggtheme = theme\_minimal())



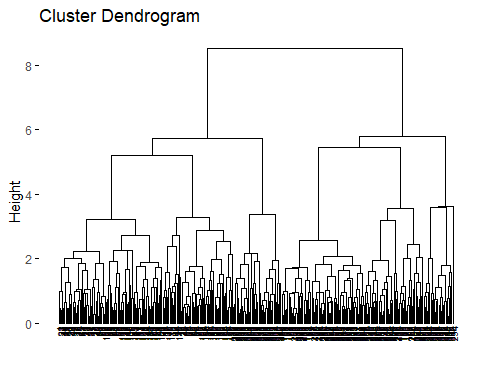
## Hierarchical clustering (DIANA)

### Part a - Dendrogram

library(factoextra)  
library(cluster)  
  
# Assuming avianmeasurements\_data is your dataset  
avianmeasurements\_scaled <- scale(avianmeasurements\_data)  
  
#### Find the dis(similarity)   
res.dist\_diana <- dist(avianmeasurements\_scaled, method = "euclidean")  
as.matrix(res.dist\_diana)[1:6, 1:6] # Distances between first 6 observations

## 1 2 3 5 6 7  
## 1 0.0000000 0.7544172 1.2467275 1.0748867 1.1639438 0.4868263  
## 2 0.7544172 0.0000000 0.9969384 1.2758963 1.6566960 0.4792699  
## 3 1.2467275 0.9969384 0.0000000 0.9751835 1.4639329 1.1346975  
## 5 1.0748867 1.2758963 0.9751835 0.0000000 0.8763334 1.2311026  
## 6 1.1639438 1.6566960 1.4639329 0.8763334 0.0000000 1.5577042  
## 7 0.4868263 0.4792699 1.1346975 1.2311026 1.5577042 0.0000000

### Conduct the HC with the diana()  
res.diana <- diana(avianmeasurements\_scaled, stand = TRUE, metric = "euclidean")  
  
#### Visualize the dendrogram  
fviz\_dend(res.diana, cex = 0.5)



### Part b - Cophentic Correlation

### Finding the cophenetic distances (height)  
res.coph\_diana <- cophenetic(res.diana)  
  
### Comparing cophenetic distances with original distances  
cor(res.dist\_diana, res.coph\_diana)

## [1] 0.8006812

The cophenetic correlation is 0.0006812, suggesting that the hierarchical clustering dendrogram provides a strong representation of the original distances in the AvianMeasurements dataset.

### Part c/d - K-value Dendrogram

The dendrogram above shows six clusters on the dendrogram. Showing the appropriate value of k would be six. The clusters are show by color below.

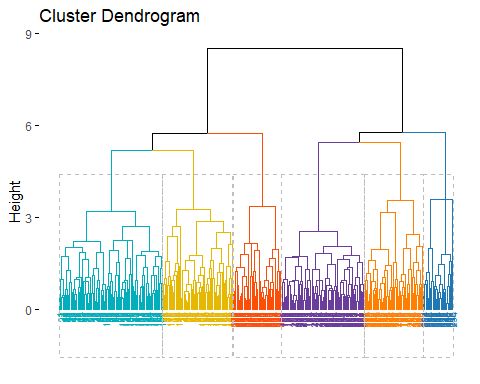
#### Cutting the dendrogram  
k\_diana <- 6 # Choose an appropriate value based on the dendrogram  
grp\_diana <- cutree(res.diana, k = k\_diana)  
  
#### Identifying groups  
head(grp\_diana)

## [1] 1 1 1 1 2 1

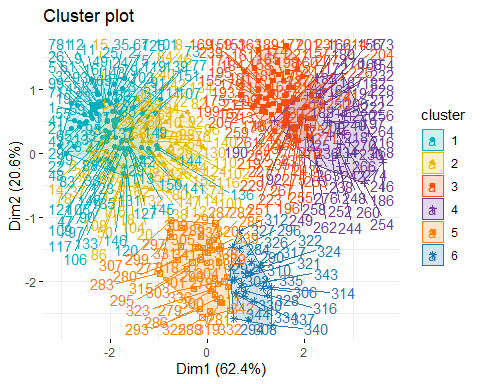
# Displaying some observations from the first group  
avianmeasurements\_data[grp\_diana == 1, ]

## X bill\_length\_mm bill\_depth\_mm flipper\_length\_mm body\_mass\_g  
## 1 1 39.1 18.7 181 3750  
## 2 2 39.5 17.4 186 3800  
## 3 3 40.3 18.0 195 3250  
## 5 5 36.7 19.3 193 3450  
## 7 7 38.9 17.8 181 3625  
## 9 9 34.1 18.1 193 3475  
## 11 11 37.8 17.1 186 3300  
## 12 12 37.8 17.3 180 3700  
## 13 13 41.1 17.6 182 3200  
## 16 16 36.6 17.8 185 3700  
## 17 17 38.7 19.0 195 3450  
## 19 19 34.4 18.4 184 3325  
## 21 21 37.8 18.3 174 3400  
## 22 22 37.7 18.7 180 3600  
## 23 23 35.9 19.2 189 3800  
## 24 24 38.2 18.1 185 3950  
## 25 25 38.8 17.2 180 3800  
## 26 26 35.3 18.9 187 3800  
## 27 27 40.6 18.6 183 3550  
## 28 28 40.5 17.9 187 3200  
## 29 29 37.9 18.6 172 3150  
## 31 31 39.5 16.7 178 3250  
## 32 32 37.2 18.1 178 3900  
## 33 33 39.5 17.8 188 3300  
## 35 35 36.4 17.0 195 3325  
## 38 38 42.2 18.5 180 3550  
## 39 39 37.6 19.3 181 3300  
## 41 41 36.5 18.0 182 3150  
## 43 43 36.0 18.5 186 3100  
## 45 45 37.0 16.9 185 3000  
## 47 47 41.1 19.0 182 3425  
## 48 48 37.5 18.9 179 2975  
## 49 49 36.0 17.9 190 3450  
## 51 51 39.6 17.7 186 3500  
## 53 53 35.0 17.9 190 3450  
## 55 55 34.5 18.1 187 2900  
## 57 57 39.0 17.5 186 3550  
## 59 59 36.5 16.6 181 2850  
## 61 61 35.7 16.9 185 3150  
## 63 63 37.6 17.0 185 3600  
## 65 65 36.4 17.1 184 2850  
## 67 67 35.5 16.2 195 3350  
## 69 69 35.9 16.6 190 3050  
## 71 71 33.5 19.0 190 3600  
## 73 73 39.6 17.2 196 3550  
## 75 75 35.5 17.5 190 3700  
## 77 77 40.9 16.8 191 3700  
## 79 79 36.2 16.1 187 3550  
## 81 81 34.6 17.2 189 3200  
## 83 83 36.7 18.8 187 3800  
## 85 85 37.3 17.8 191 3350  
## 88 88 36.9 18.6 189 3500  
## 90 90 38.9 18.8 190 3600  
## 91 91 35.7 18.0 202 3550  
## 93 93 34.0 17.1 185 3400  
## 95 95 36.2 17.3 187 3300  
## 97 97 38.1 18.6 190 3700  
## 99 99 33.1 16.1 178 2900  
## 101 101 35.0 17.9 192 3725  
## 103 103 37.7 16.0 183 3075  
## 105 105 37.9 18.6 193 2925  
## 106 106 39.7 18.9 184 3550  
## 107 107 38.6 17.2 199 3750  
## 109 109 38.1 17.0 181 3175  
## 111 111 38.1 16.5 198 3825  
## 113 113 39.7 17.7 193 3200  
## 117 117 38.6 17.0 188 2900  
## 119 119 35.7 17.0 189 3350  
## 120 120 41.1 18.6 189 3325  
## 121 121 36.2 17.2 187 3150  
## 123 123 40.2 17.0 176 3450  
## 125 125 35.2 15.9 186 3050  
## 127 127 38.8 17.6 191 3275  
## 129 129 39.0 17.1 191 3050  
## 131 131 38.5 17.9 190 3325  
## 133 133 36.8 18.5 193 3500  
## 135 135 38.1 17.6 187 3425  
## 136 136 41.1 17.5 190 3900  
## 137 137 35.6 17.5 191 3175  
## 139 139 37.0 16.5 185 3400  
## 141 141 40.2 17.1 193 3400  
## 142 142 40.6 17.2 187 3475  
## 143 143 32.1 15.5 188 3050  
## 144 144 40.7 17.0 190 3725  
## 145 145 37.3 16.8 192 3000  
## 146 146 39.0 18.7 185 3650  
## 148 148 36.6 18.4 184 3475  
## 149 149 36.0 17.8 195 3450  
## 150 150 37.8 18.1 193 3750  
## 151 151 36.0 17.1 187 3700

#### Plotting using color with six distinct colors  
fviz\_dend(res.diana, k = k\_diana, cex = 0.5,   
 k\_colors = c("#00AFBB", "#E7B800", "#FC4E07", "#6A3D9A", "#FF7F00", "#1F78B4"),   
 color\_labels\_by\_k = TRUE, rect = TRUE)



### Visualizing results as a scatter plot  
fviz\_cluster(list(data = avianmeasurements\_scaled, cluster = grp\_diana),   
 palette = c("#00AFBB", "#E7B800", "#FC4E07", "#6A3D9A", "#FF7F00", "#1F78B4"),  
 ellipse.type = "convex", repel = TRUE, show.clust.cent = FALSE,   
 ggtheme = theme\_minimal())



### AGNES vs DIANA

The results of this dendrogram is slightly vary from the first dendrogram. AGNES and DIANA suggest different cluster structures for the AvianMeasurements dataset. AGNES proposes three clusters, while DIANA suggests six clusters. However the clustering using AGNES seems to fit the data more appropriately, as the clusters using DIANA seem to be unclear on the scatterplot.

## Clustering Summary

In conclusion a k-value of 3 is most optimal for the data set AvianMeasurements. The k-value of 3 is supported by two analysis where the clusters are clear and distinct. The data also shows that AGNES clustering is preferred because the cophentic correlation is higher.

# Part 2 - Classification using Bagging, Boosting and RF

## Test and Train

library(caret)

## Warning: package 'caret' was built under R version 4.3.2

## Loading required package: lattice

library(rpart)  
library(ipred)

## Warning: package 'ipred' was built under R version 4.3.2

library(dplyr)

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

abalone <- read.csv("abalone.csv")  
colnames(abalone) <- c("Sex", "Length", "Diameter", "Height", "WholeWeight", "ShuckedWeight", "VisceraWeight", "ShellWeight", "Rings")  
set.seed(123)  
  
  
  
# Create an index for splitting the data (70/30)  
split\_index <- createDataPartition(abalone$Sex, p = 0.7, list = FALSE)  
  
# Create the training set  
abalone\_train <- abalone[split\_index, ]  
  
# Create the test set  
abalone\_test <- abalone[-split\_index, ]

## Bagging

library(caret)  
library(ggplot2)  
  
#Change characters to numbers  
abalone\_train$Sex <- factor(abalone\_train$Sex)  
abalone\_test$Sex <- factor(abalone\_test$Sex)  
# Train a bagging model  
abalone\_bagging <- bagging(formula = Sex ~ ., data = abalone\_train, mfinal = 10,  
 control = rpart.control(maxdepth = 1))  
  
#Predictions on the test set  
abalone\_pred <- predict(abalone\_bagging, newdata = abalone\_test)  
  
#Model using a confusion matrix  
conf\_matrix <- table(abalone\_pred, abalone\_test$Sex, dnn = c("Predicted Class", "Observed Class"))  
conf\_matrix

## Observed Class  
## Predicted Class F I M  
## F 0 0 0  
## I 72 302 90  
## M 320 100 368

#Accuracy is .1445687  
# Calculate accuracy  
accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)  
accuracy

## [1] 0.5351438

##Boosting

library(adabag)

## Warning: package 'adabag' was built under R version 4.3.2

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 4.3.2

## Loading required package: doParallel

## Warning: package 'doParallel' was built under R version 4.3.2

## Loading required package: iterators

## Warning: package 'iterators' was built under R version 4.3.2

## Loading required package: parallel

##   
## Attaching package: 'adabag'

## The following object is masked from 'package:ipred':  
##   
## bagging

library(caret)  
  
abalone\_train$Sex <- factor(abalone\_train$Sex)  
abalone\_test$Sex <- factor(abalone\_test$Sex)  
  
# Train an AdaBoost model  
abalone\_adaboost <- boosting(Sex ~ ., data = abalone\_train, mfinal = 10)  
  
#Predictions on the test set  
abalone\_pred <- predict(abalone\_adaboost, newdata = abalone\_test)  
  
  
#Model using a confusion matrix  
conf\_matrix <- table(abalone\_pred$class, abalone\_test$Sex, dnn = c("Predicted Class", "Observed Class"))  
conf\_matrix

## Observed Class  
## Predicted Class F I M  
## F 127 24 117  
## I 68 315 106  
## M 197 63 235

#Accuracy is .5463259  
accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)  
accuracy

## [1] 0.5407348

## Random Forest

library(randomForest)

## Warning: package 'randomForest' was built under R version 4.3.2

## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## The following object is masked from 'package:ggplot2':  
##   
## margin

library(caret)  
  
abalone\_train$Sex <- factor(abalone\_train$Sex)  
abalone\_test$Sex <- factor(abalone\_test$Sex)  
  
#Train Random Forest model  
abalone\_rf <- randomForest(Sex ~ ., data = abalone\_train, ntree = 100)  
  
#Predictions on the test set  
abalone\_pred <- predict(abalone\_rf, newdata = abalone\_test)  
  
#Model using a confusion matrix  
conf\_matrix <- table(abalone\_pred, abalone\_test$Sex, dnn = c("Predicted Class", "Observed Class"))  
conf\_matrix

## Observed Class  
## Predicted Class F I M  
## F 164 35 150  
## I 50 310 82  
## M 178 57 226

#Accuracy is 0.5463259  
accuracy <- sum(diag(conf\_matrix)) / sum(conf\_matrix)  
accuracy

## [1] 0.5591054

## Comparison

In comparing the three ensemble learning models—Bagging, Boosting, and Random Forest—applied to the Abalone dataset, distinct characteristics and performance metrics emerge. Random Forest exhibited the highest accuracy among the models, reaching 54.63%, followed closely by Boosting with the same accuracy and Bagging with a considerably lower accuracy of 14.46%. Random Forest and Bagging demonstrated resilience to noisy data and tended to be less prone to overfitting due to their ensemble nature, with Random Forest showing the highest robustness.