DANA 4840 Project - Partition Clustering

Aryan Mukherjee, Maryam Gadimova, Patricia Tating, Roman Shrestha

1. Research Statement on Breast Cancer Dataset

Breast cancer is a critical health issue, with early and accurate detection playing a vital role in treatment and patient outcomes. This dataset captures the features of cell nuclei through comprehensive measurements taken during breast cancer biopsies. Each observation spans several measurements and includes characteristics like radius, texture, perimeter, area, and others. Additionally, labels describing the tumor's malignancy or benignity are included in the dataset.

K-Means and Partitioning Around Medoids (PAM) clustering methods will be used to segment the dataset into clusters, validating the tumor's diagnosis of either malignant or benign cases. This clustering analysis not only provides insights into the heterogeneity of breast cancer but also aids in identifying key features that distinguish between benign and malignant cases. The findings can contribute to improving diagnostic accuracy and personalized treatment approaches.

2. Preliminaries

Before diving into the cluster analysis, let's first thoroughly examine and understand our data. This preliminary step will allow us to identify key patterns and characteristics within the dataset, ensuring a solid foundation for accurate analysis. By doing so, we can address any potential data quality issues and refine our approach for more meaningful results.

```r
library("tidyverse")
library("factoextra")
library("dendextend")
library("hopkins")
library("corrplot")
library("cluster")
library("patchwork")
library("clValid")
library("EMCluster")
```

2.1. Reading the Data

```r
wdbc <- read.table("data/wdbc.csv", header = T, sep = ",")
head(wdbc)
```

```
##         id diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842302         M       17.99        10.38         122.80    1001.0
## 2   842517         M       20.57        17.77         132.90    1326.0
## 3 84300903         M       19.69        21.25         130.00    1203.0
## 4 84348301         M       11.42        20.38          77.58     386.1
## 5 84358402         M       20.29        14.34         135.10    1297.0
## 6   843786         M       12.45        15.70          82.57     477.1
##   smoothness_mean compactness_mean concavity_mean concave.points_mean
## 1         0.11840          0.27760         0.3001             0.14710
## 2         0.08474          0.07864         0.0869             0.07017
## 3         0.10960          0.15990         0.1974             0.12790
## 4         0.14250          0.28390         0.2414             0.10520
## 5         0.10030          0.13280         0.1980             0.10430
## 6         0.12780          0.17000         0.1578             0.08089
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.2419                0.07871    1.0950     0.9053        8.589
## 2        0.1812                0.05667    0.5435     0.7339        3.398
## 3        0.2069                0.05999    0.7456     0.7869        4.585
## 4        0.2597                0.09744    0.4956     1.1560        3.445
## 5        0.1809                0.05883    0.7572     0.7813        5.438
## 6        0.2087                0.07613    0.3345     0.8902        2.217
##   area_se smoothness_se compactness_se concavity_se concave.points_se
## 1  153.40      0.006399        0.04904      0.05373           0.01587
## 2   74.08      0.005225        0.01308      0.01860           0.01340
## 3   94.03      0.006150        0.04006      0.03832           0.02058
## 4   27.23      0.009110        0.07458      0.05661           0.01867
## 5   94.44      0.011490        0.02461      0.05688           0.01885
## 6   27.19      0.007510        0.03345      0.03672           0.01137
##   symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
## 1     0.03003             0.006193        25.38         17.33          184.60
## 2     0.01389             0.003532        24.99         23.41          158.80
## 3     0.02250             0.004571        23.57         25.53          152.50
## 4     0.05963             0.009208        14.91         26.50           98.87
## 5     0.01756             0.005115        22.54         16.67          152.20
## 6     0.02165             0.005082        15.47         23.75          103.40
##   area_worst smoothness_worst compactness_worst concavity_worst
## 1     2019.0           0.1622            0.6656          0.7119
## 2     1956.0           0.1238            0.1866          0.2416
## 3     1709.0           0.1444            0.4245          0.4504
## 4      567.7           0.2098            0.8663          0.6869
## 5     1575.0           0.1374            0.2050          0.4000
## 6      741.6           0.1791            0.5249          0.5355
##   concave.points_worst symmetry_worst fractal_dimension_worst
## 1               0.2654         0.4601                 0.11890
## 2               0.1860         0.2750                 0.08902
## 3               0.2430         0.3613                 0.08758
## 4               0.2575         0.6638                 0.17300
## 5               0.1625         0.2364                 0.07678
## 6               0.1741         0.3985                 0.12440
```

2.1.1. Checking Data Structure

```
dim(wdbc)
```

```
## [1] 569  32
```

```r
str(wdbc)
```

```
## 'data.frame':    569 obs. of  32 variables:
##  $ id                     : int  842302 842517 84300903 84348301 84358402 843786 844359
##  $ diagnosis              : chr  "M" "M" "M" "M" ...
##  $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
##  $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
##  $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
##  $ area_mean              : num  1001 1326 1203 386 1297 ...
##  $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
##  $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
##  $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
##  $ concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
##  $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
##  $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
##  $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
##  $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
##  $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
##  $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
##  $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
##  $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
##  $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
##  $ concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
##  $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
##  $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
##  $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
##  $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
##  $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
##  $ area_worst             : num  2019 1956 1709 568 1575 ...
##  $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
##  $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
##  $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
##  $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
##  $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
##  $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
```

We can see that our data comprises 569 instances of breast cancer biopsies and 32 features related to cell nuclei characteristics all of which are numerical variables except for 'Diagnosis' which is a target variable and 'ID' which is a unique identifier.

2.2. Feature Explanation

The 'wdbc' dataset includes 32 features as detailed below:

- 'ID' (identifier) - patient ID
- 'Diagnosis' (categorical) - Diagnosis of breast tissues (M = Malignant, B = Benign)
- 'radius_mean' (numerical) - Mean of distances from center to points on the perimeter
- 'texture_mean' (numerical) - Standard deviation of gray-scale values
- 'perimeter_mean'(numerical) - Mean size of the core tumor
- 'area_mean'(numerical) - Mean area of the tumor cells
- 'smoothness_mean' (numerical) - Mean of local variation in radius lengths
- 'compactness_mean' (numerical) - Mean of perimeter^2 / area - 1.0

- ‘concavity_mean’ (numerical) - Mean of severity of concave portions of the contour
- ‘concave_points_mean’ (numerical) - Mean for number of concave portions of the contour
- ‘symmetry_mean’ (numerical) - Mean symmetry of the tumor cells
- ‘fractal_dimension_mean’ (numerical) - Mean “coastline approximation” of the tumor cells
- ‘radius_se’ (numerical) - Standard error of the radius of the tumor cells
- ‘texture_se’ (numerical) - Standard error of the texture of the tumor cells
- ‘perimeter_se’ (numerical) - Standard error of the perimeter of the tumor cells
- ‘area_se’ (numerical) - Standard error of the area of the tumor cells
- ‘smoothness_se’ (numerical) - Standard error of the smoothness of the tumor cells
- ‘compactness_se’ (numerical) - Standard error of the compactness of the tumor cells
- ‘concavity_se’ (numerical) - Standard error of the concavity of the tumor cells
- ‘concave_points_se’ (numerical) - Standard error of the number of concave portions of the contour of the tumor cells
- ‘symmetry_se’ (numerical) - Standard error of the symmetry of the tumor cells
- ‘fractal_dimension_se’ (numerical) - Standard error of the “coastline approximation” of the tumor cells
- ‘radius_worst’(numerical) - Worst (largest) radius of the tumor cells
- ‘texture_worst’ (numerical) - Worst (most severe) texture of the tumor cells
- ‘perimeter_worst’ (numerical) - Worst (largest) perimeter of the tumor cells
- ‘area_worst’ (numerical) - Worst (largest) area of the tumor cells
- ‘smoothness_worst’ (numerical) - Worst (most severe) smoothness of the tumor cells
- ‘compactness_worst’ (numerical) - Worst (most severe) compactness of the tumor cells
- ‘concavity_worst’ (numerical) - Worst (most severe) concavity of the tumor cells
- ‘concave_points_worst’ (numerical) - Worst (most severe) number of concave portions of the contour of the tumor cells
- ‘symmetry_worst’ (numerical) - Worst (most severe) symmetry of the tumor cells
- ‘fractal_dimension_worst’ (numerical) - Worst (most severe) “coastline approximation” of the tumor cells

2.3. Exploratory Data Analysis

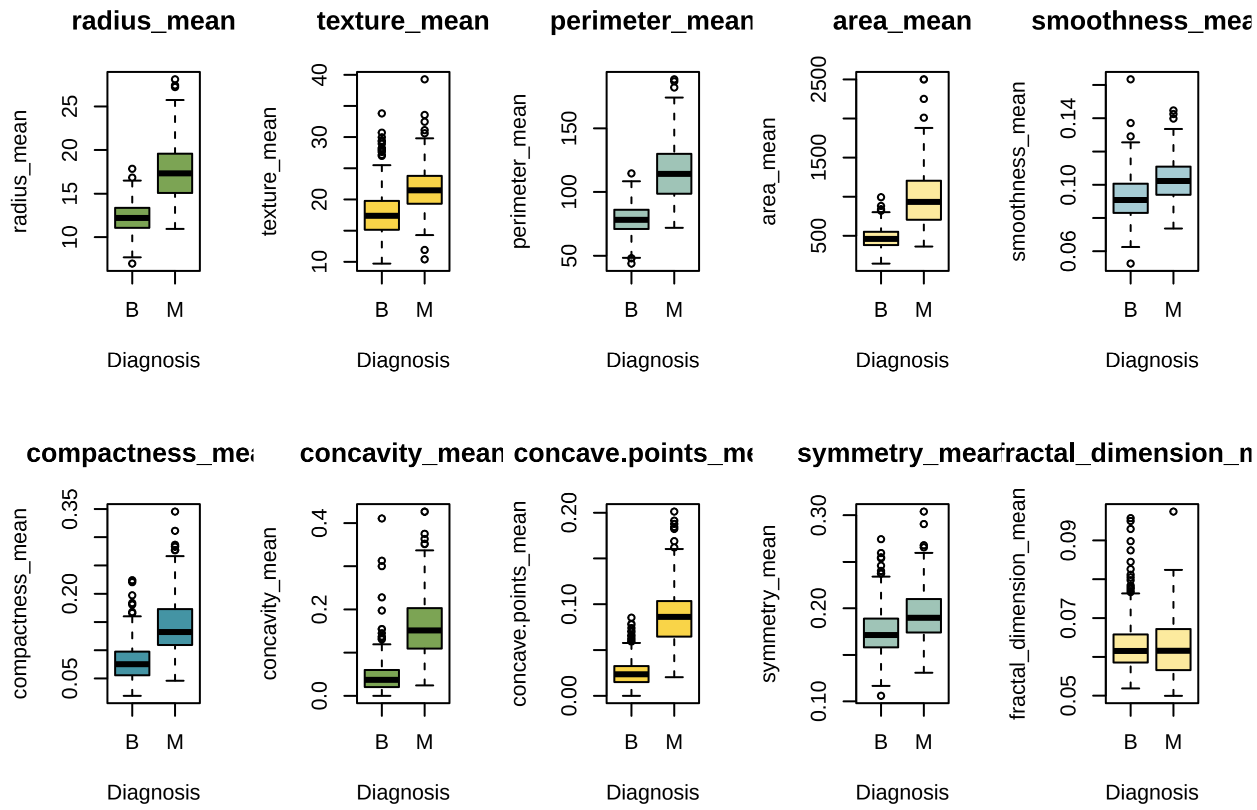
2.3.1. Checking Missing Values

```r
missing_wdbc <- sapply(wdbc, function(x) sum(is.na(x)))
missing_wdbc
```

```
##                     id              diagnosis            radius_mean 
##                      0                      0                      0 
##           texture_mean         perimeter_mean              area_mean 
##                      0                      0                      0 
##        smoothness_mean       compactness_mean         concavity_mean 
##                      0                      0                      0 
##    concave.points_mean          symmetry_mean fractal_dimension_mean 
##                      0                      0                      0 
##              radius_se             texture_se           perimeter_se 
##                      0                      0                      0 
##                area_se          smoothness_se         compactness_se 
##                      0                      0                      0 
##           concavity_se      concave.points_se            symmetry_se 
##                      0                      0                      0 
##   fractal_dimension_se           radius_worst          texture_worst 
##                      0                      0                      0 
##         perimeter_worst              area_worst       smoothness_worst
##                       0                       0                      0
##       compactness_worst         concavity_worst   concave.points_worst
##                       0                       0                      0
##          symmetry_worst fractal_dimension_worst
##                       0                       0
```

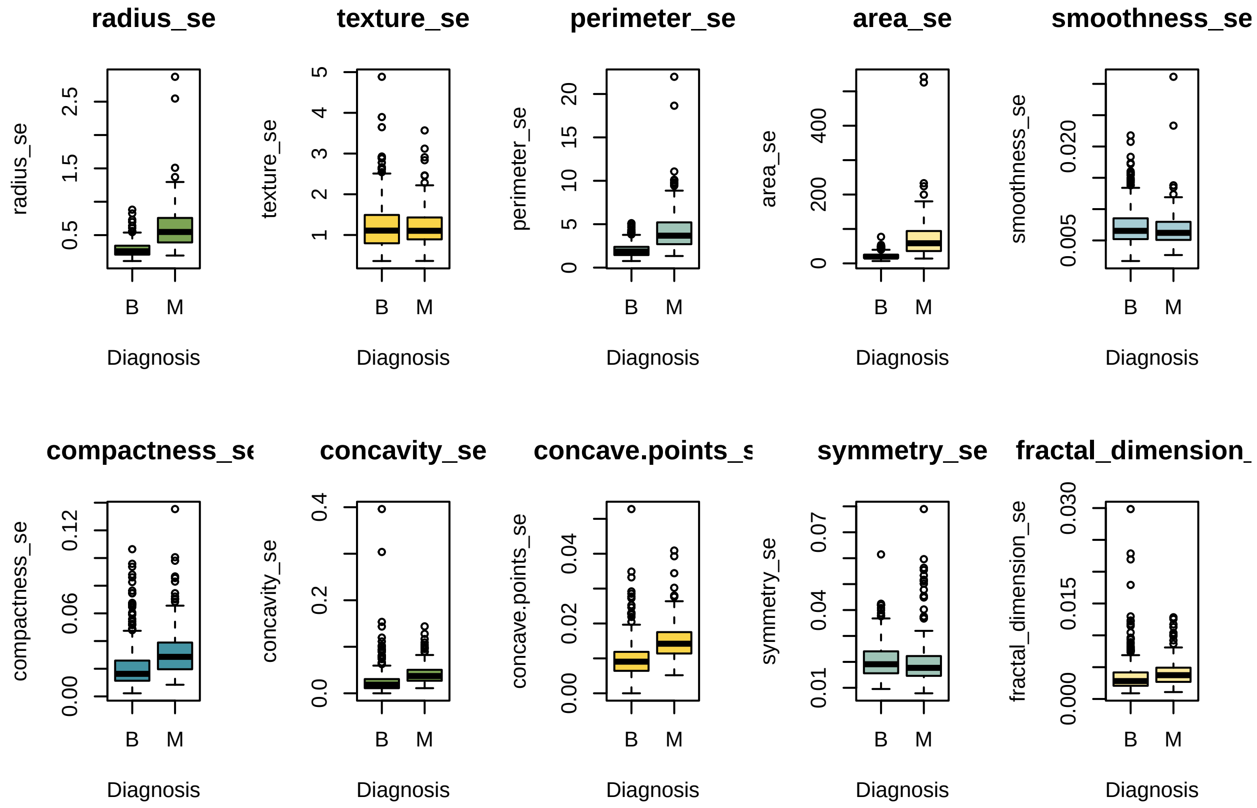
2.3.2. Boxplots for Different Feature Groups

```
color_palette <- c("#4494a4", "#7ca454", "#f9d448", "#9fc4b7", "#fcea9e", "#a6ccd4")

par(mfrow = c(2, 5))

mean_columns <- grep("_mean", names(wdbc), value = TRUE)

for (i in seq_along(mean_columns)) {
  column_name <- mean_columns[i]

  boxplot(wdbc[[column_name]] ~ wdbc$diagnosis,
          xlab = "Diagnosis",
          ylab = column_name,
          main = paste(column_name),
          col = color_palette[i %% length(color_palette) + 1])
}
```



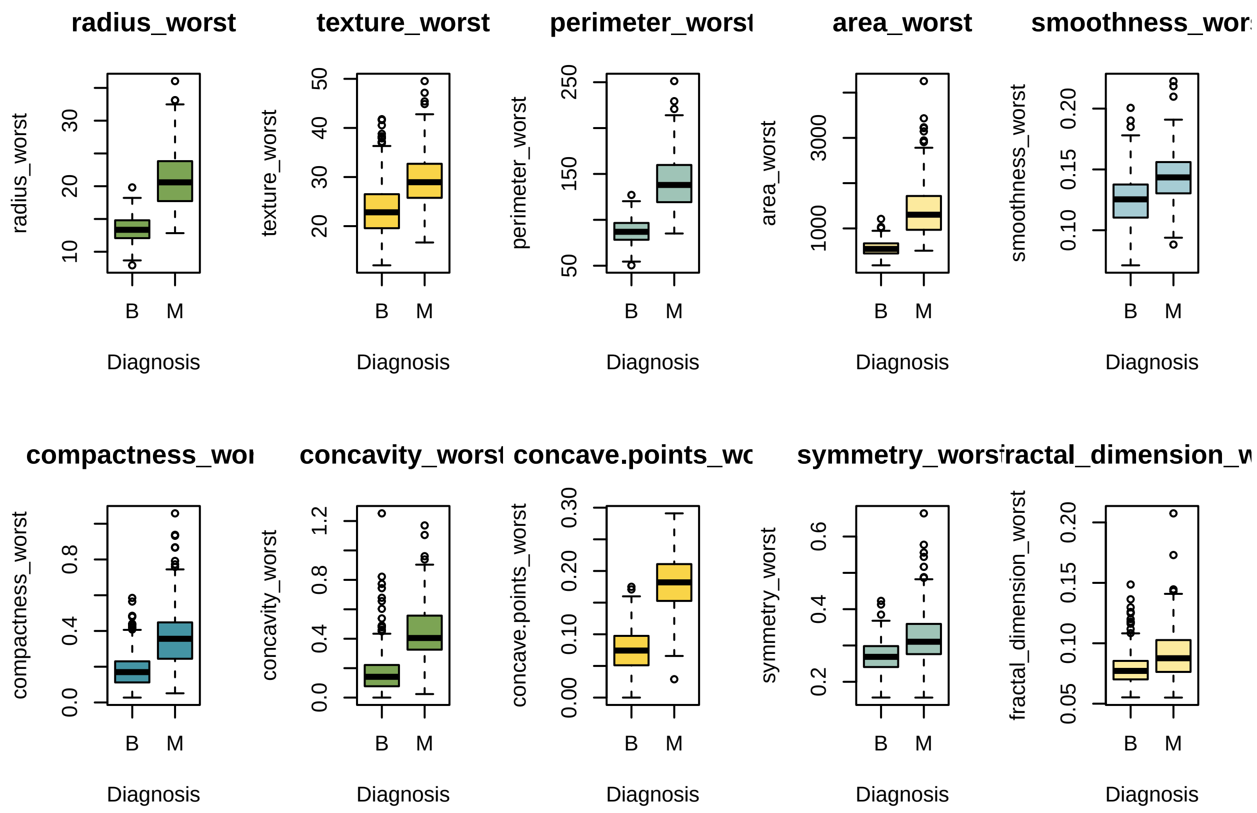
```r
par(mfrow = c(1, 1))
```

```r
par(mfrow = c(2, 5))

se_columns <- grep("_se", names(wdbc), value = TRUE)

for (i in seq_along(se_columns)) {
  column_name <- se_columns[i]

  boxplot(wdbc[[column_name]] ~ wdbc$diagnosis,
          xlab = "Diagnosis",
          ylab = column_name,
          main = paste(column_name),
          col = color_palette[i %% length(color_palette) + 1])
}
```



```r
par(mfrow = c(1, 1))
```

```r
par(mfrow = c(2, 5))

worst_columns <- grep("_worst", names(wdbc), value = TRUE)

for (i in seq_along(worst_columns)) {
  column_name <- worst_columns[i]
  boxplot(wdbc[[column_name]] ~ wdbc$diagnosis,
          xlab = "Diagnosis",
          ylab = column_name,
          main = paste(column_name),
          col = color_palette[i %% length(color_palette) + 1])
}
```



```
par(mfrow = c(1, 1))
```

The malignant (M) diagnosis consistently exhibits higher medians and wider ranges across several features specifically in mean and worst values of the cell nuclei characteristics, indicating that M diagnosis forms a distinct cluster characterized by these statistics.

Several outliers are observed across the features; however, given the clinical nature of the data, the outliers have been retained, as they likely represent natural variations rather than measurement errors.

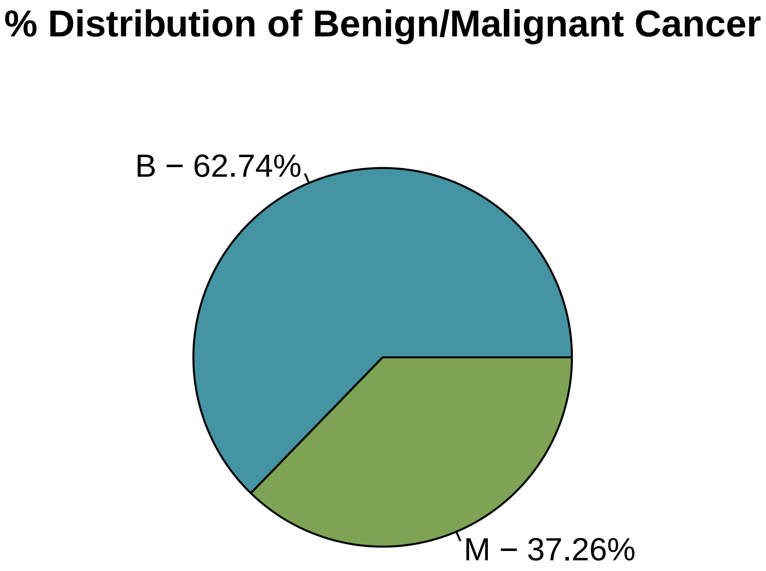
2.3.3. Pie Chart for Diagnosis Distribution

```
diagnosis_freq <- table(wdbc$diagnosis)
diagnosis_rel_freq <- prop.table(diagnosis_freq) * 100
diagnosis_rel_freq
```

```
##
```

```
##        B        M
## 62.74165 37.25835
```

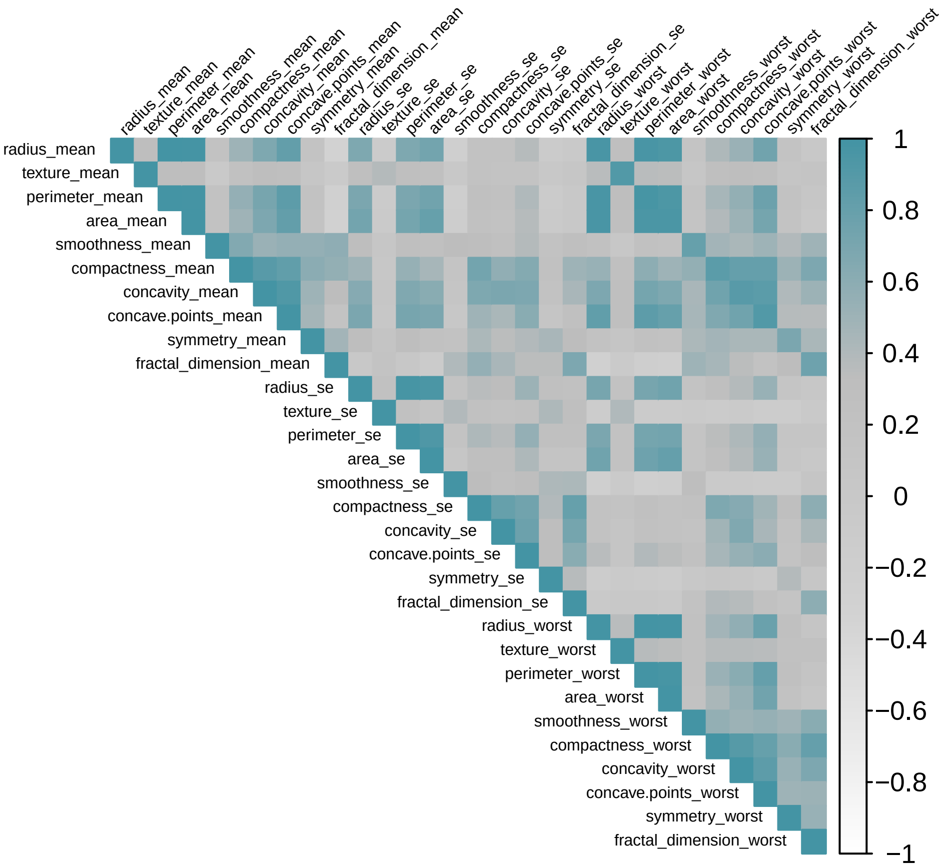
```
pie(diagnosis_rel_freq,
    main = "% Distribution of Benign/Malignant Cancer",
    labels = c("B - 62.74%", "M - 37.26%"),
    col = color_palette)
```



Benign cancer makes up approximately 62.9% of the dataset, while Malignant cancer constitutes about 37.1%. Benign cancer cases are approximately 1.69 times more prevalent than Malignant cancer cases in the dataset.

2.3.4. Correlation Heatmap

```
library(corrplot)
wdbc_numerical <- wdbc[, -c(1, 2)]
cor_matrix <- cor(wdbc_numerical, use = "complete.obs")

corrplot(cor_matrix,
         method = "color",
         type = "upper",
         col = colorRampPalette(c("white","lightgrey", "grey", "#4494a4"))(200),
         tl.col = "black",
         tl.srt = 45,
         tl.cex = 0.5)
```



We can clearly see that there exists multicollinearity between features in the dataset.

2.4. Data Pre-processing

```
diagnosis <- wdbc$diagnosis

wdbc_scaled <- data.frame(scale(wdbc_numerical))
rownames(wdbc_scaled) <- wdbc$id
wdbc <- wdbc_scaled
head(wdbc)
```

```
##          radius_mean texture_mean perimeter_mean  area_mean smoothness_mean
## 842302     1.0960995   -2.0715123      1.2688173  0.9835095       1.5670875
## 842517     1.8282120   -0.3533215      1.6844726  1.9070303      -0.8262354
## 84300903   1.5784992    0.4557859      1.5651260  1.5575132       0.9413821
## 84348301  -0.7682333    0.2535091     -0.5921661 -0.7637917       3.2806668
## 84358402   1.7487579   -1.1508038      1.7750113  1.8246238       0.2801253
## 843786    -0.4759559   -0.8346009     -0.3868077 -0.5052059       2.2354545
##          compactness_mean concavity_mean concave.points_mean symmetry_mean
## 842302          3.2806281     2.65054179           2.5302489   2.215565542
## 842517         -0.4866435    -0.02382489           0.5476623   0.001391139
## 84300903        1.0519999     1.36227979           2.0354398   0.938858720
## 84348301        3.3999174     1.91421287           1.4504311   2.864862154
## 84358402        0.5388663     1.36980615           1.4272370  -0.009552062
## 843786          1.2432416     0.86554001           0.8239307   1.004517928
##          fractal_dimension_mean  radius_se texture_se perimeter_se    area_se
## 842302                2.2537638  2.4875451 -0.5647681    2.8305403  2.4853907
## 842517               -0.8678888  0.4988157 -0.8754733    0.2630955  0.7417493
## 84300903             -0.3976580  1.2275958 -0.7793976    0.8501802  1.1802975
## 84348301              4.9066020  0.3260865 -0.1103120    0.2863415 -0.2881246
## 84358402             -0.5619555  1.2694258 -0.7895490    1.2720701  1.1893103
## 843786                1.8883435 -0.2548461 -0.5921406   -0.3210217 -0.2890039
##          smoothness_se compactness_se concavity_se concave.points_se
## 842302      -0.2138135     1.31570389    0.7233897        0.66023900
## 842517      -0.6048187    -0.69231710   -0.4403926        0.25993335
## 84300903    -0.2967439     0.81425704    0.2128891        1.42357487
## 84348301     0.6890953     2.74186785    0.8187979        1.11402678
## 84358402     1.4817634    -0.04847723    0.8277425        1.14319885
## 843786       0.1562093     0.44515196    0.1598845       -0.06906279
##          symmetry_se fractal_dimension_se radius_worst texture_worst
## 842302     1.1477468           0.90628565    1.8850310   -1.35809849
## 842517    -0.8047423          -0.09935632    1.8043398   -0.36887865
## 84300903   0.2368272           0.29330133    1.5105411   -0.02395331
## 84348301   4.7285198           2.04571087   -0.2812170    0.13386631
## 84358402  -0.3607748           0.49888916    1.2974336   -1.46548091
## 843786     0.1340009           0.48641784   -0.1653528   -0.31356043
##          perimeter_worst area_worst smoothness_worst compactness_worst
## 842302         2.3015755  1.9994782        1.3065367         2.6143647
## 842517         1.5337764  1.8888270       -0.3752817        -0.4300658
## 84300903       1.3462906  1.4550043        0.5269438         1.0819801
## 84348301      -0.2497196 -0.5495377        3.3912907         3.8899747
## 84358402       1.3373627  1.2196511        0.2203623        -0.3131190
## 843786        -0.1149083 -0.2441054        2.0467119         1.7201029
##          concavity_worst concave.points_worst symmetry_worst
## 842302         2.1076718            2.2940576      2.7482041
## 842517        -0.1466200            1.0861286     -0.2436753
## 84300903       0.8542223            1.9532817      1.1512420
## 84348301       1.9878392            2.1738732      6.0407261
## 84358402       0.6126397            0.7286181     -0.8675896
## 843786         1.2621327            0.9050914      1.7525273
##          fractal_dimension_worst
## 842302                 1.9353117
## 842517                 0.2809428
## 84300903               0.2012142
## 84348301               4.9306719
## 84358402              -0.3967505
## 843786                 2.2398308
```

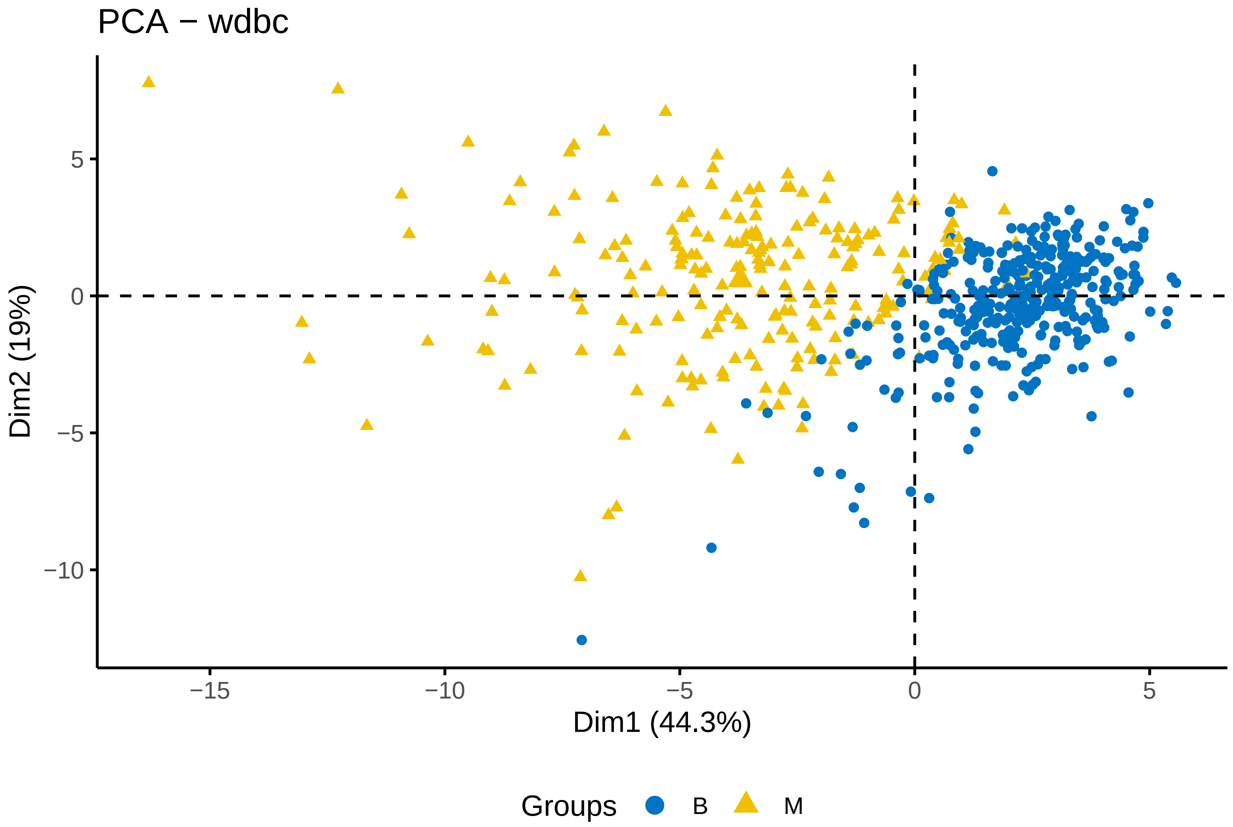
Our features have different scale of measurements, so we standardized the data to ensure each variable contributes equally to the distance calculations, preventing variables with larger scales to have more weight in the clustering results.

3. Pre-clustering Assessment

Before performing clustering analysis, it is crucial to conduct a pre-clustering assessment to evaluate the dataset's cluster tendency and determine the optimal clustering approach. Tools like the Hopkins statistic and VAT can help assess whether the data points possess significant clustering tendencies. Once cluster tendency is established, the next step involves finding the optimal number of clusters. This can be achieved using methods such as the Elbow Method, Silhouette Analysis, or the Gap Statistic, each providing insights into the most meaningful way to partition the data.

3.1. Assessing Cluster Tendency

```
fviz_pca_ind(
  prcomp(wdbc),
  title = "PCA - wdbc",
  habillage = diagnosis,
  palette = "jco",
  geom = "point",
  ggtheme = theme_classic(),
  legend = "bottom"
)
```



When visualizing our data, we can clearly see how our Benign and Malignant groups are clustered together. However, we have to validate this clustering.

3.1.1. Hopkins Statistics

```
set.seed(69)
```

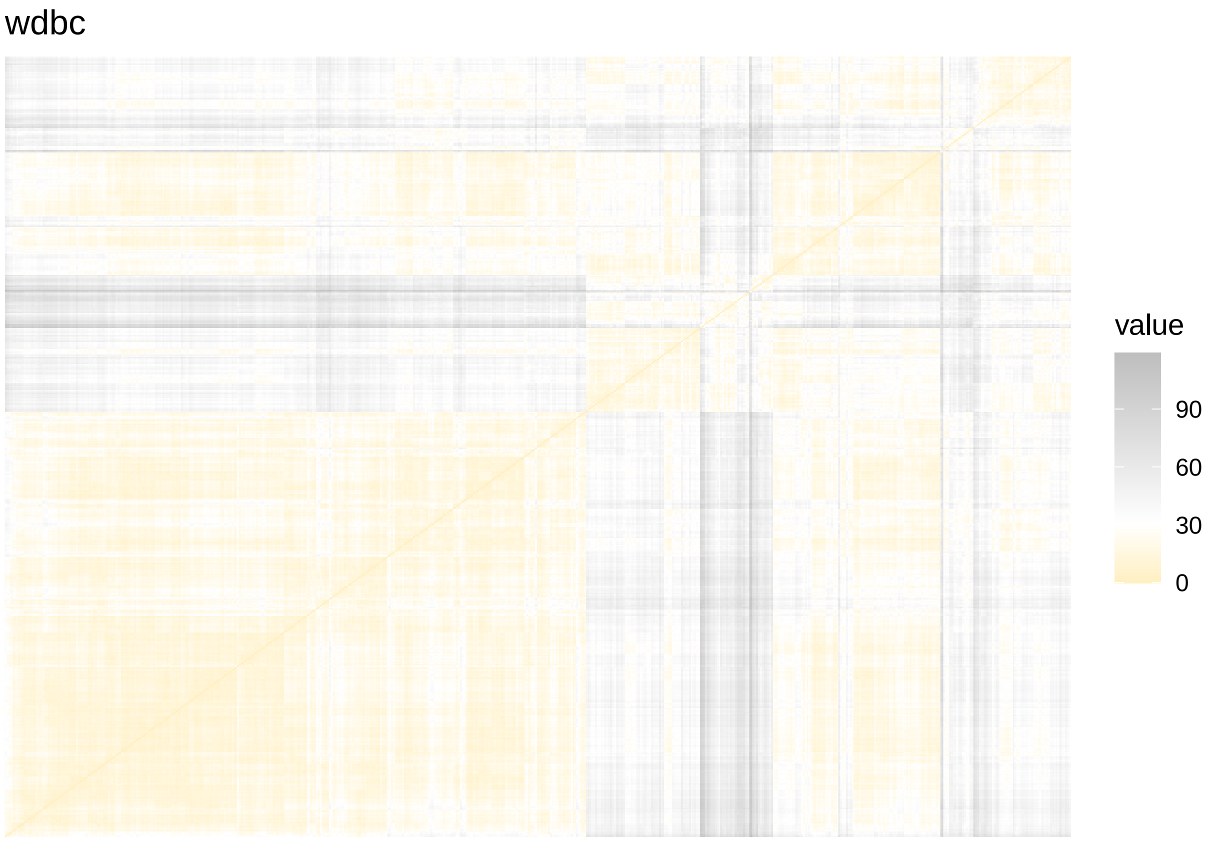
```r
hopkins_wdbc <- hopkins(wdbc, m = ceiling(nrow(wdbc) / 10))
hopkins_wdbc
```

```
## [1] 0.9999997
```

A Hopkins statistic value of 0.9999997 indicates that the dataset exhibits a high degree of clusterability.

3.1.2. Visual Assessment of Cluster Tendency (VAT)

```r
fviz_dist(
  dist(wdbc, method = "manhattan"),
  show_labels = FALSE,
  gradient = list(low = "#f9d448", mid = "white", high = "grey")
) + labs(title = "wdbc")
```

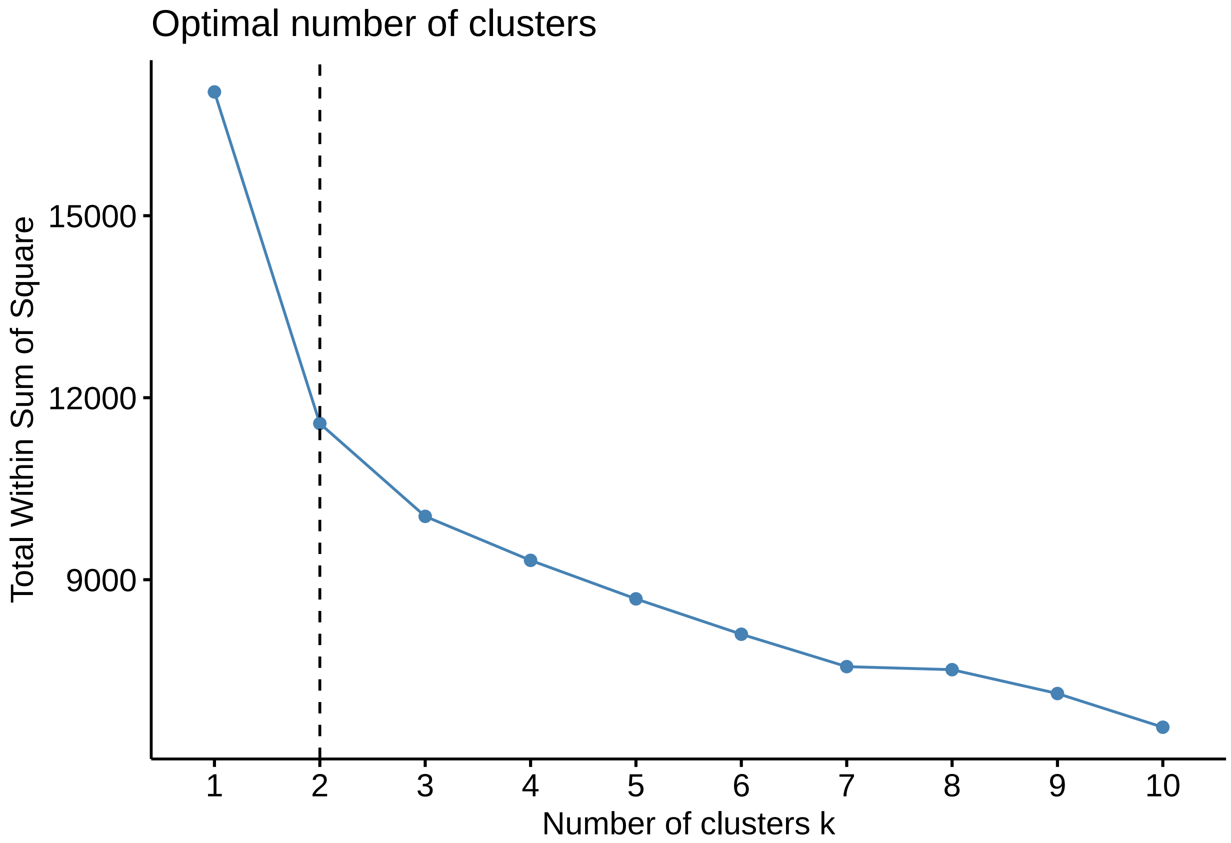


Based on the visual assessment and the Hopkins statistic of 0.9999997, the breast cancer dataset is confirmed to be suitable for clustering. Before proceeding with the partitioning clustering analysis, it is essential to determine the optimal number of clusters.

3.2. Finding the Optimal Number of Clusters

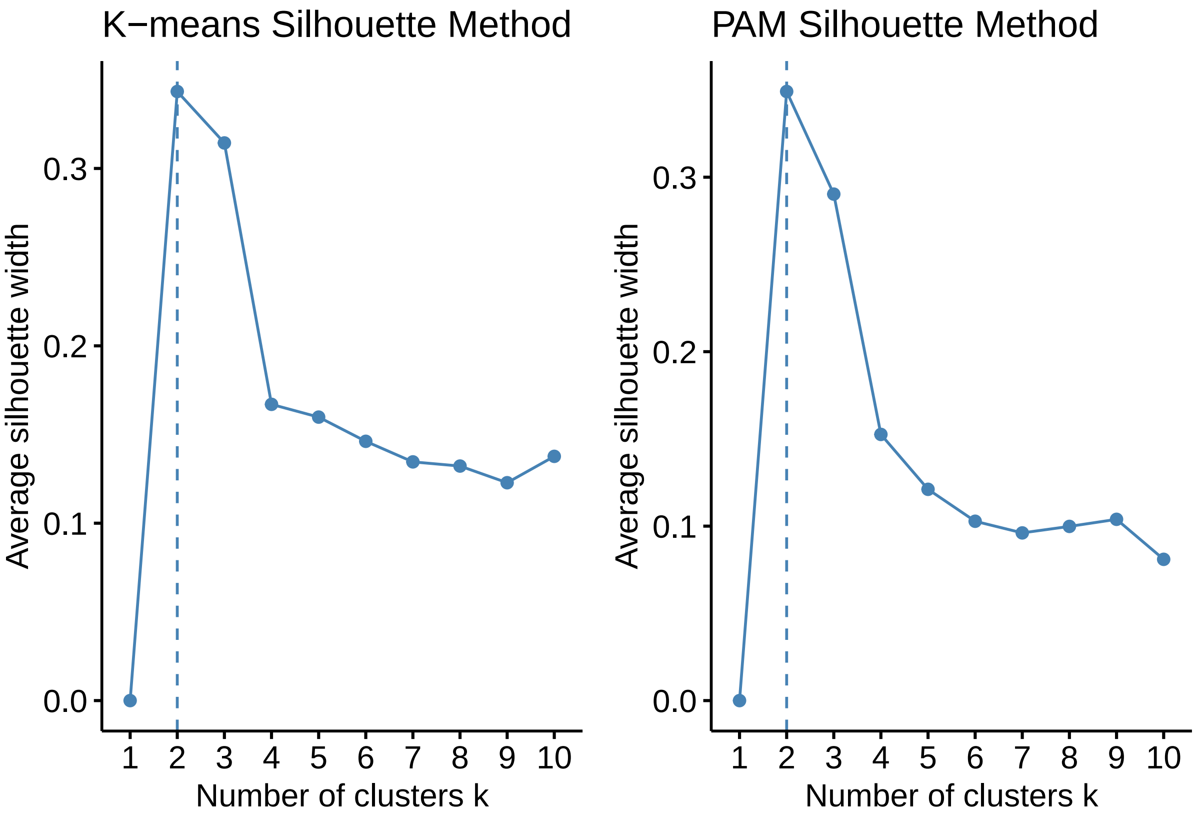
3.2.1. Elbow Method

```
wdbc_elbow_kmeans <- fviz_nbclust(wdbc, kmeans, method = "wss") +
  geom_vline(xintercept = 2, linetype = 2)
wdbc_elbow_kmeans
```



3.2.2. Silhouette Method

```
wdbc_silhouette_kmeans <- fviz_nbclust(wdbc, kmeans, method = "silhouette") +
  labs(title = "K-means Silhouette Method")

wdbc_silhouette_pam <- fviz_nbclust(wdbc, pam, method = "silhouette") +
  labs(title = "PAM Silhouette Method")

wdbc_silhouette_kmeans +
  wdbc_silhouette_pam +
  plot_layout(ncol = 2)
```



3.2.3. Gap Statistics

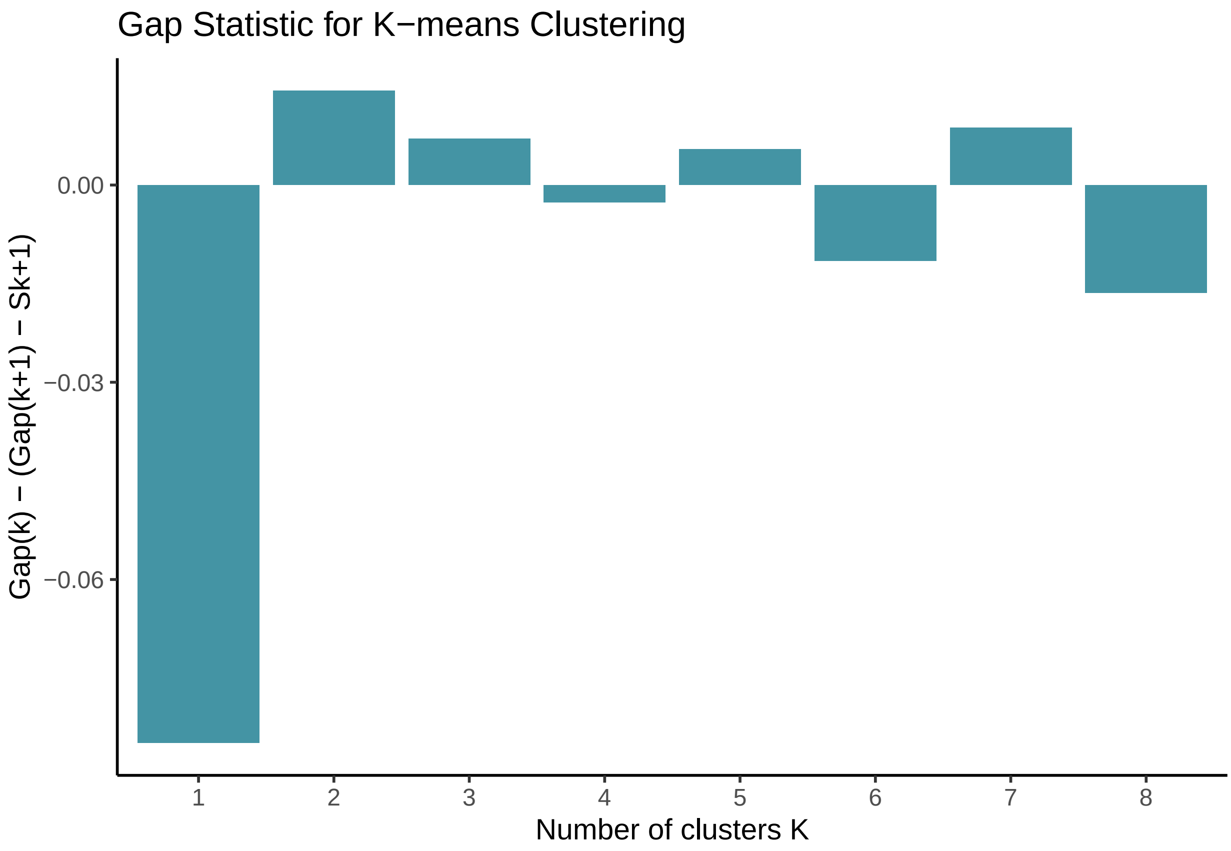
Below, we calculate the gap statistics for the k-means clustering of the breast cancer dataset:

```
get_cluster_diff <- function(gap_stat, max_k = 10) {
  gap_df <- as.data.frame(gap_stat$Tab)

  gap_diff_list <- vector()
  gap_val_list <- gap_df$gap
  s_val_list <- gap_df$SE.sim

  for (k in 1:max_k) {
    if (k < max_k - 1) {
      val <- gap_val_list[k] -
        (gap_val_list[k + 1] -
          s_val_list[k + 1])

      gap_diff_list <- append(gap_diff_list, val)
    }
  }

  return(gap_diff_list)
}

max_k <- 10
```

```
gap_stat <- clusGap(wdbc, kmeans, K.max = max_k, B = 500)
```

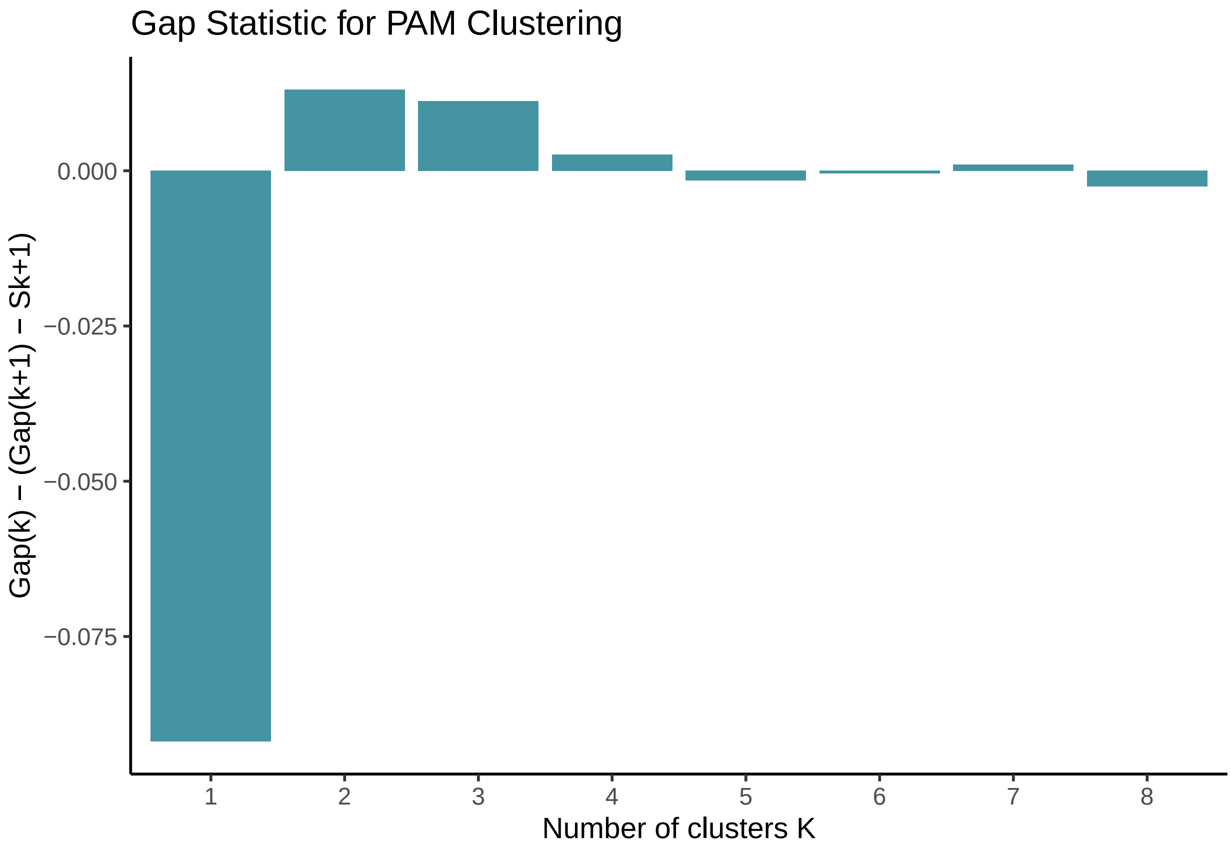
```
## Warning: did not converge in 10 iterations
```

```
gap_diff_list <- get_cluster_diff(gap_stat, max_k)
pos_neg_df <- data.frame(cluster = factor(seq_along(gap_diff_list)),
                         gap_diff = gap_diff_list)

kmeans_gap <- ggplot(data = pos_neg_df, aes(x = cluster, y = gap_diff)) +
  geom_bar(stat = "identity", fill = "#4494a4") +
  xlab("Number of clusters K") +
  ylab("Gap(k) - (Gap(k+1) - Sk+1)") +
  ggtitle("Gap Statistic for K-means Clustering") +
  theme_classic()
kmeans_gap
```

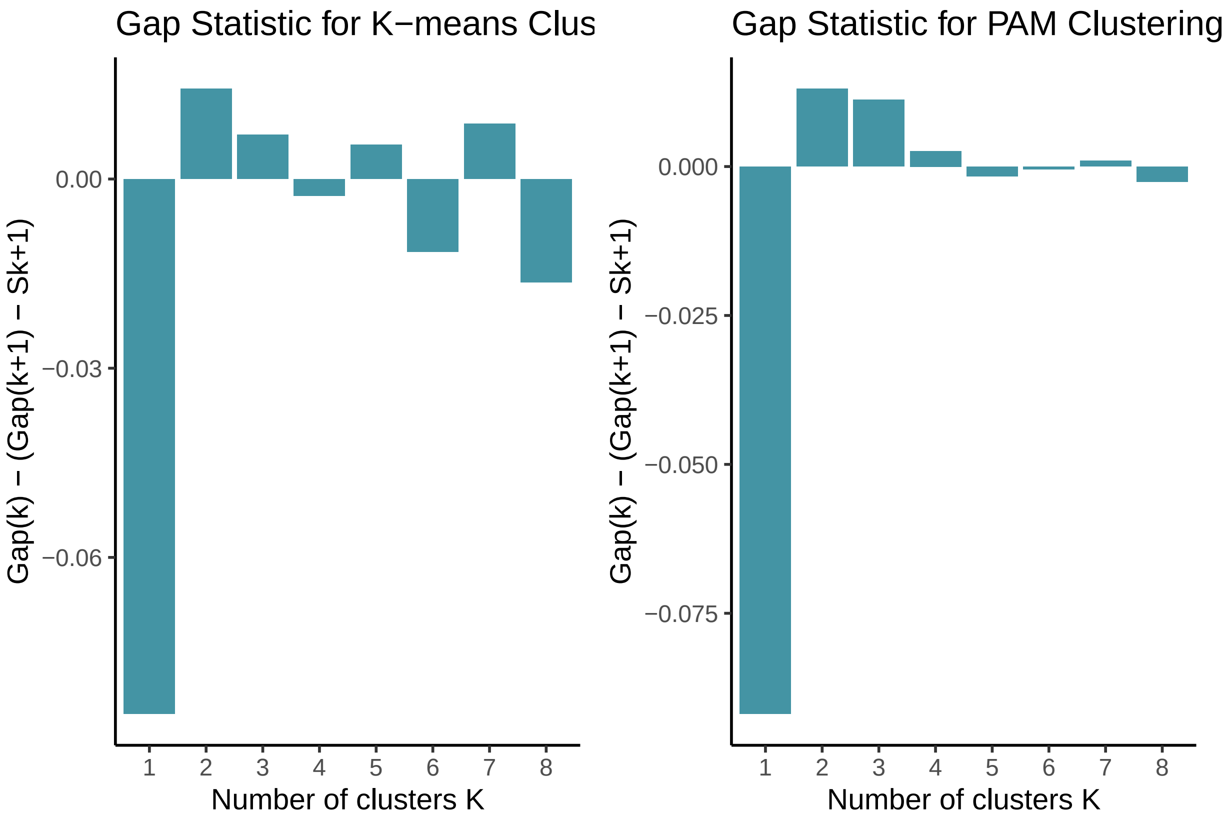


Below, we calculate the gap statistics for the PAM clustering of the breast cancer dataset:

```
max_k <- 10
gap_stat <- clusGap(wdbc, pam, K.max = max_k, B = 500)
gap_diff_list <- get_cluster_diff(gap_stat, max_k)
pos_neg_df <- data.frame(cluster = factor(seq_along(gap_diff_list)),
                         gap_diff = gap_diff_list)

pam_gap <- ggplot(data = pos_neg_df, aes(x = cluster, y = gap_diff)) +
  geom_bar(stat = "identity", fill = "#4494a4") +
  xlab("Number of clusters K") +
  ylab("Gap(k) - (Gap(k+1) - Sk+1)") +
  ggtitle("Gap Statistic for PAM Clustering") +
  theme_classic()
pam_gap
```



```
kmeans_gap +
  pam_gap +
  plot_layout(ncol = 2)
```

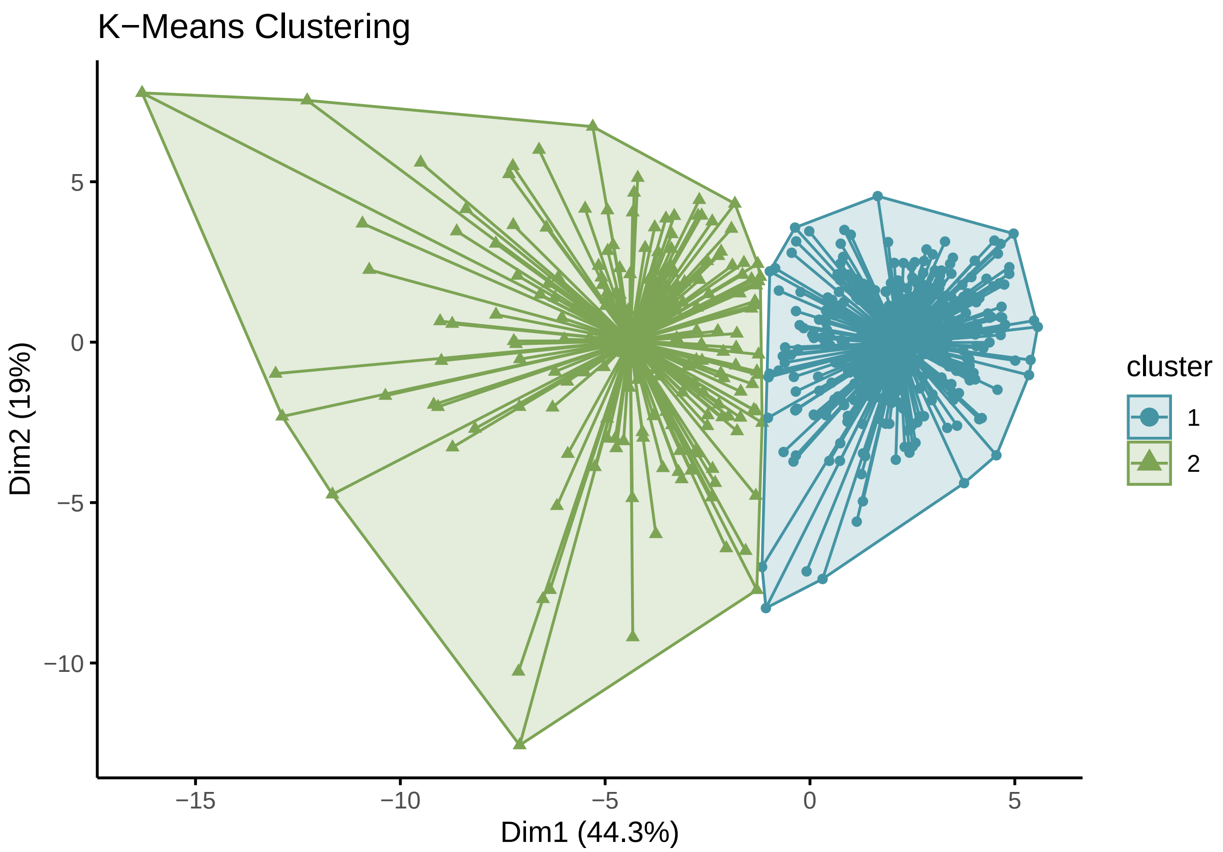


Both graphs show gap statistics turning positive in K=2. Overall, the elbow, silhouette, and gap statistics methods all suggest K=2 as the optimal number of clusters.

4. Clustering Analysis

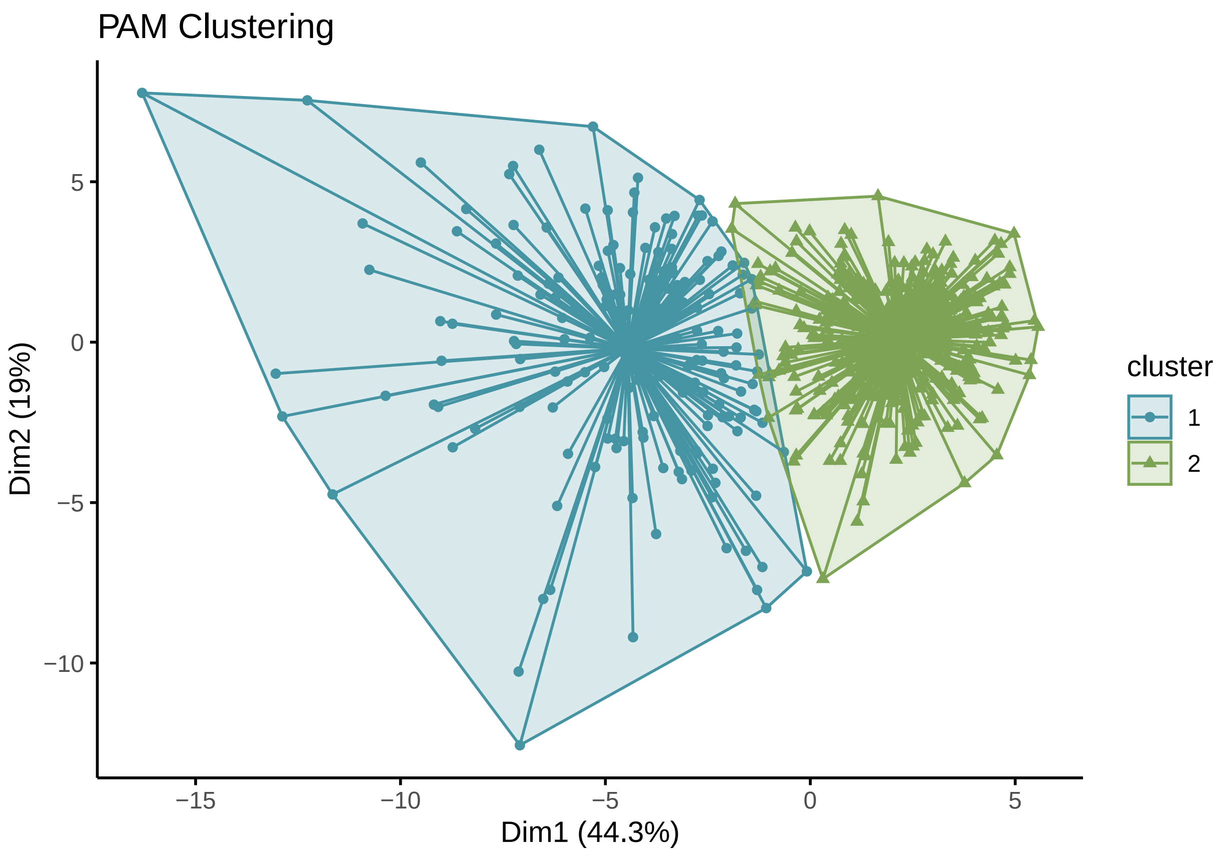
4.1. K-Means Clustering

```r
set.seed(101)

km.res <- kmeans(wdbc, centers = 2, nstart = 100)

kmeans_graph <- fviz_cluster(
  km.res,
  data = wdbc,
  palette = c("#4494a4", "#7ca454"),
  ellipse.type = "convex",
  star.plot = TRUE,
  ellipse = TRUE,
  geom = "point",
  main = "K-Means Clustering",
  ggtheme = theme_classic()
)
kmeans_graph
```



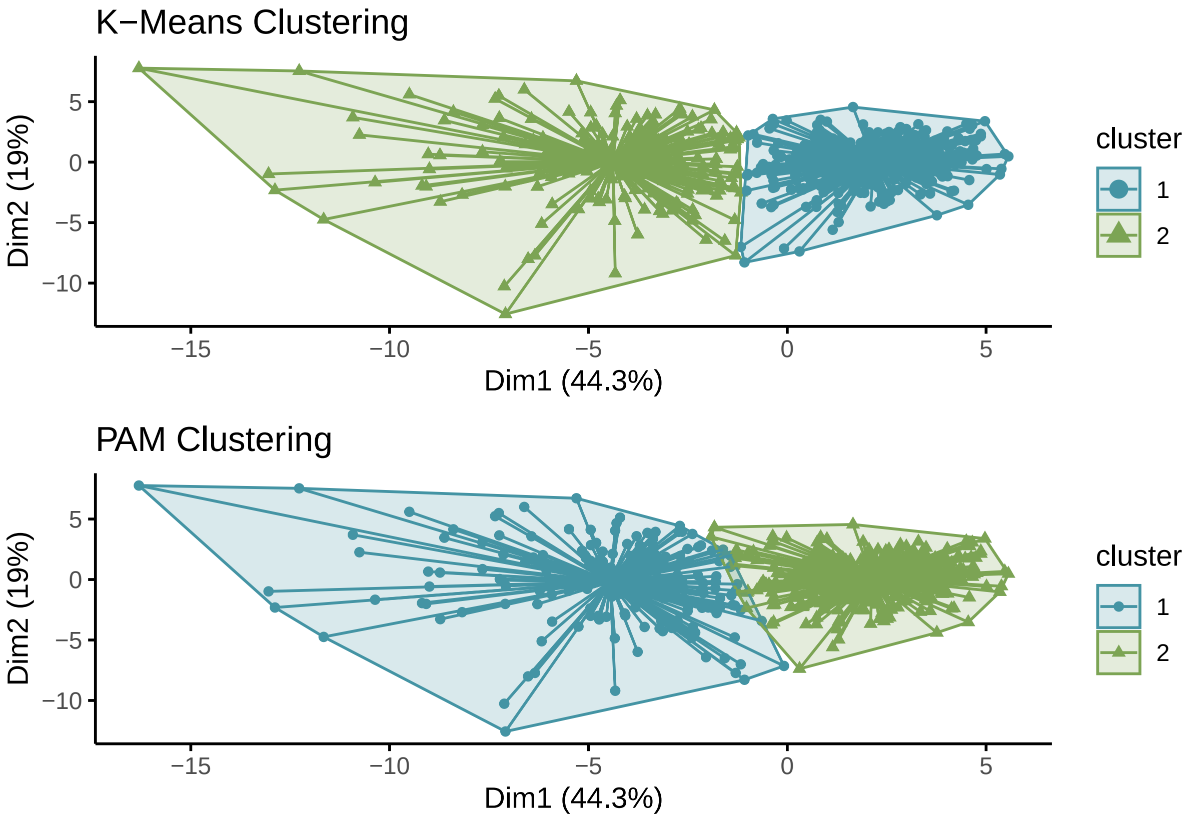
4.2. Partition Around Medoid (PAM) Clustering

```
set.seed(101)

pam.res <- pam(wdbc, k = 2)

pam_graph <- fviz_cluster(
  pam.res,
  data = wdbc,
  palette = c("#4494a4", "#7ca454"),
  ellipse.type = "convex",
  star.plot = TRUE,
  ellipse = TRUE,
  geom = "point",
  main = "PAM Clustering",
  ggtheme = theme_classic()
)
pam_graph
```



4.3. Comparing K-Means and PAM

```
kmeans_graph + pam_graph + plot_layout(ncol = 1)
```



We aim for clusters that are compact and well-separated. Upon visual inspection of our plots, we observed that the PAM clusters exhibit slight overlap, indicating that they are not as distinct and well-separated as the clusters formed by K-Means. To verify these observations, we will employ various cluster validation techniques.

5. Cluster Validation

5.1. External Validation

5.1.1. Contingency Table - Diagnosis vs. Cluster Results

```
## Creating a data frame with diagnosis, k-means and PAM cluster results
encoded_diagnosis <- ifelse(diagnosis == "M", 1, 2)

wdbc_results <- cbind(
  wdbc,
  diagnosis = encoded_diagnosis,
  kmeans_cluster = ifelse(km.res$cluster == 1, 2, 1),
  pam_cluster = pam.res$clustering
)
```

```
kmeans_contingency_table <- table(wdbc_results$diagnosis, wdbc_results$kmeans_cluster)
kmeans_contingency_table
```

```
##
##       1   2
##   1 175  37
##   2  14 343
```

```
pam_contingency_table <- table(wdbc_results$diagnosis, wdbc_results$pam_cluster)
pam_contingency_table
```

```
##
##       1   2
##   1 167  45
##   2  17 340
```

The contingency table shows that the K-Means method has a 8.96% misclassification rate compared to the ground truth variable, representing the actual diagnosis. In comparison, the PAM method has a slightly higher misclassification rate of 10.90%.

5.1.2. Rand Index

```
kmeans_rand <- RRand(wdbc_results$diagnosis, wdbc_results$kmeans_cluster)
kmeans_rand
```

```
##    Rand adjRand  Eindex
##  0.8365  0.6707  0.5897
```

```
pam_rand <- RRand(wdbc_results$diagnosis, wdbc_results$pam_cluster)
pam_rand
```

```
##    Rand adjRand  Eindex
##  0.8055  0.6079  0.5961
```

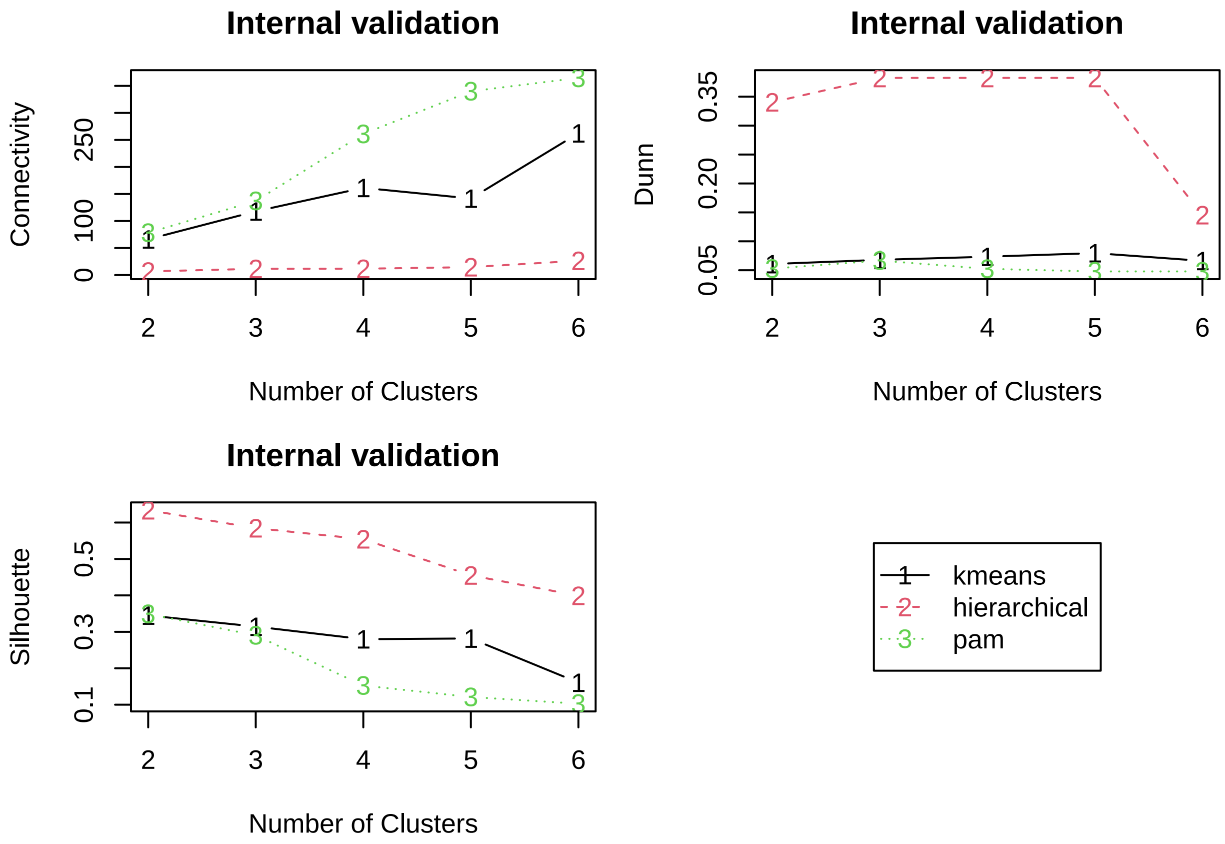
The Rand Index (RI) for K-Means clustering is 0.8365, indicating a strong alignment between the clustering results and the actual diagnosis. For the PAM method, the Rand Index is slightly lower at 0.8055, but still reflects a good agreement with the actual diagnosis, though not as high as with K-Means.

5.2. Internal Validation

```
intern_wdbc <- clValid(
  wdbc,
  2:6,
  clMethods = c("kmeans", "hierarchical", "pam"),
  validation = "internal"
)

summary(intern_wdbc)
```

```
##
## Clustering Methods:
##  kmeans hierarchical pam
##
## Cluster sizes:
##  2 3 4 5 6
##
## Validation Measures:
##                                   2        3        4        5        6
##
## kmeans       Connectivity   66.2083 117.8448 161.3952 141.6123 262.4095
##              Dunn            0.0608   0.0680   0.0734   0.0797   0.0662
##              Silhouette      0.3450   0.3144   0.2798   0.2818   0.1607
## hierarchical Connectivity    6.7202  11.5782  11.7448  14.6738  26.4615
##              Dunn            0.3405   0.3825   0.3825   0.3825   0.1454
##              Silhouette      0.6340   0.5846   0.5543   0.4550   0.3991
## pam          Connectivity   78.2040 137.0464 261.0294 339.9877 364.7885
##              Dunn            0.0528   0.0667   0.0525   0.0479   0.0479
##              Silhouette      0.3491   0.2903   0.1526   0.1211   0.1028
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.7202 hierarchical 2
## Dunn         0.3825 hierarchical 3
## Silhouette   0.6340 hierarchical 2
```

```
par(mfrow = c(2, 2), mar = c(4, 4, 3, 1))

plot(intern_wdbc, legend = FALSE)

plot(
  nClusters(intern_wdbc),
  measures(intern_wdbc, "Dunn")[, , 1],
  type = "n",
  axes = FALSE,
  xlab = "",
  ylab = ""
)
legend("center", clusterMethods(intern_wdbc), col = 1:9, lty = 1:9, pch = paste(1:9))
```



- **Connectivity:** The optimal score of 6.7202 is achieved with hierarchical clustering at 2 clusters. This indicates the most compact clustering with minimal inter-cluster distances is in k=2.
- **Dunn Index:** The highest Dunn index of 0.3825 is observed with hierarchical clustering at 3 clusters, suggesting the best separation between clusters.
- **Silhouette Width:** The maximum silhouette width of 0.6340 is found with hierarchical clustering at 2 clusters, reflecting the highest average similarity within clusters and dissimilarity between clusters.

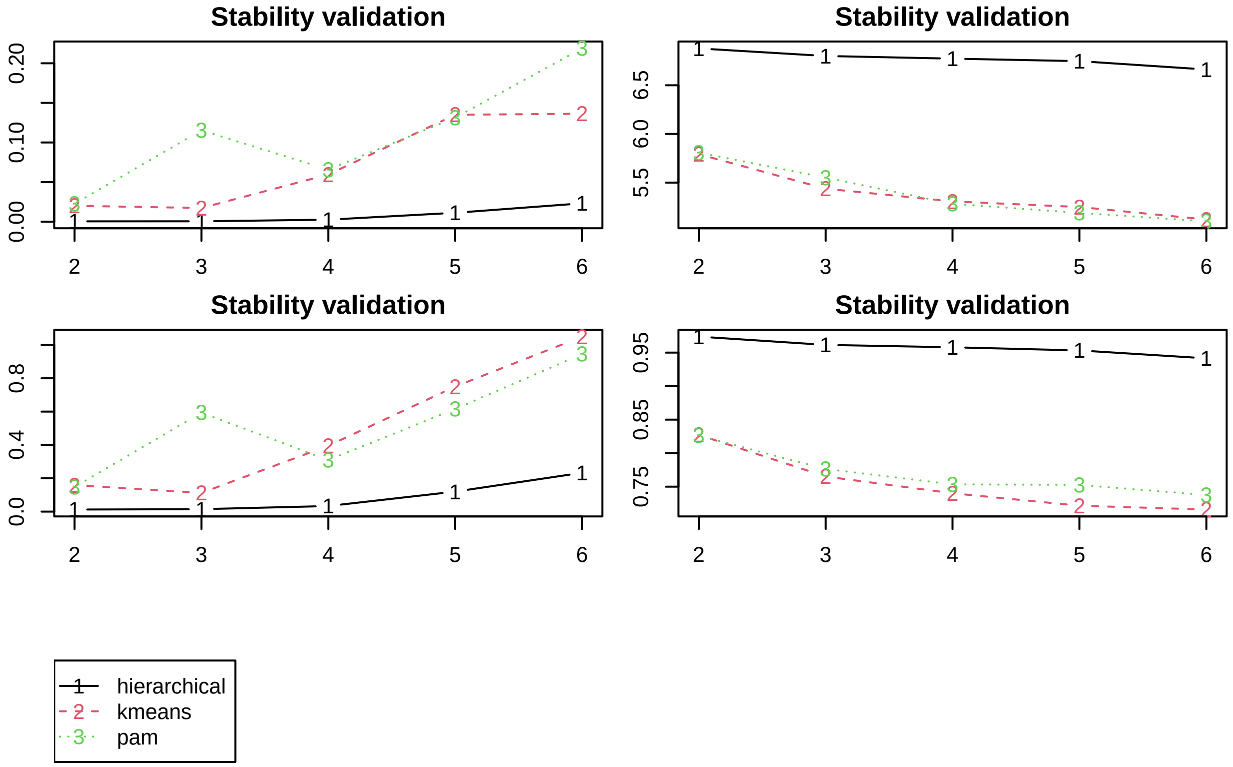
Hierarchical clustering shows the best performance based on all three metrics, suggesting well-defined and separated clusters. K-Means also performs relatively well but shows a decline in cluster quality as the number of clusters increases. PAM, while providing some separation, consistently underperforms compared to the other methods.

5.3. Stability Validation

```
stab_wdbc <- clValid(
  wdbc,
  nClust = 2:6,
  clMethods = c("hierarchical", "kmeans", "pam"),
  validation = "stability"
)

optimal_scores_stab <- optimalScores(stab_wdbc)
optimal_scores_stab
```

```
##              Score       Method Clusters
## APN 0.0003905487 hierarchical        2
## AD  5.1021325696          pam        6
## ADM 0.0124118166 hierarchical        2
## FOM 0.7159036749       kmeans        6
```

```r
par(mfrow = c(3, 2), mar = c(2, 2, 2, 1))

plot(stab_wdbc, measure = c("APN", "AD", "ADM", "FOM"), legend = FALSE)

plot(
  nClusters(stab_wdbc),
  measures(stab_wdbc, "APN")[, , 1],
  type = "n",
  axes = FALSE,
  xlab = "",
  ylab = ""
)
legend("left", clusterMethods(stab_wdbc), col = 1:9, lty = 1:9, pch = paste(1:9))
```



- **APN (Average Path Length):** The lowest APN of 0.0003905487 is achieved with hierarchical clustering at 3 clusters, indicating minimal average path lengths among clusters.
- **AD (Average Distance):** The highest AD of 5.1021325696 is found with PAM clustering at 6 clusters, reflecting the average distance within clusters.

- **ADM (Average Dissimilarity):** The lowest ADM of 0.0124118166 is achieved with hierarchical clustering at 3 clusters, showing minimal average dissimilarity within clusters.
- **FOM (Freeman's Measure):** The highest FOM of 0.7159036749 is observed with PAM clustering at 6 clusters, suggesting better clustering performance according to Freeman's measure.
- **Hierarchical Clustering with 3 Clusters:** Optimal for APN and ADM, indicating compact and well-defined clusters with minimal average path length and dissimilarity.
- **PAM Clustering with 6 Clusters:** Optimal for AD and FOM, reflecting more spread-out clusters and superior overall clustering performance based on Freeman's Measure.

6. Conclusion and Recommendation

6.1. Conclusion

In this study, we utilized partitioning clustering techniques, specifically K-Means and Partitioning Around Medoids (PAM), to analyze the breast cancer dataset. The analysis aimed to distinguish between benign and malignant cases based on the clustering of various tumor cell features.

Our results demonstrated that K-Means clustering provided a slightly better alignment with the actual diagnosis labels compared to PAM.

The optimal number of clusters was determined to be two, based on various validation methods such as the Elbow Method, Silhouette Analysis, and Gap Statistics. This result aligns with the ground truth variable or the known diagnosis between benign and malignant tumor.

These findings can contribute to improving diagnostic accuracy and personalized treatment approaches by identifying key features that differentiate between benign and malignant cases, thereby aiding in early detection and targeted therapies.

6.2. Recommendation

1. **Advanced Clustering Techniques:** Explore more advanced clustering techniques such as hierarchical clustering or model-based clustering, which may provide deeper insights into the data structure. We've seen on the internal validation that hierarchical performs better.
2. **Further Feature Analysis:** Future research should focus on the significance of individual features and their contributions to the clustering process. This can help in identifying key biomarkers for early detection and treatment planning.
3. **Integration with Clinical Data:** Integrating these clustering results with clinical data such as patient history, treatment outcomes, and genetic information could provide a more holistic understanding of breast cancer subtypes and their respective treatment strategies.