# DANA4840 Project - Partitional Clustering

M.Gadimova, A.Mukherjee, R.Shrestha, P.Tating

2024-07-29

## 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.

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

set.seed(101)
```

### 2.1. Reading the Data

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

```
##    X842302 M X17.99 X10.38 X122.8  X1001 X0.1184 X0.2776 X0.3001 X0.1471
## 1   842517 M  20.57  17.77 132.90 1326.0 0.08474 0.07864  0.0869 0.07017
## 2 84300903 M  19.69  21.25 130.00 1203.0 0.10960 0.15990  0.1974 0.12790
## 3 84348301 M  11.42  20.38  77.58  386.1 0.14250 0.28390  0.2414 0.10520
## 4 84358402 M  20.29  14.34 135.10 1297.0 0.10030 0.13280  0.1980 0.10430
## 5   843786 M  12.45  15.70  82.57  477.1 0.12780 0.17000  0.1578 0.08089
## 6   844359 M  18.25  19.98 119.60 1040.0 0.09463 0.10900  0.1127 0.07400
##   X0.2419 X0.07871 X1.095 X0.9053 X8.589 X153.4 X0.006399 X0.04904 X0.05373
## 1  0.1812  0.05667 0.5435  0.7339  3.398  74.08  0.005225  0.01308  0.01860
## 2  0.2069  0.05999 0.7456  0.7869  4.585  94.03  0.006150  0.04006  0.03832
## 3  0.2597  0.09744 0.4956  1.1560  3.445  27.23  0.009110  0.07458  0.05661
## 4  0.1809  0.05883 0.7572  0.7813  5.438  94.44  0.011490  0.02461  0.05688
## 5  0.2087  0.07613 0.3345  0.8902  2.217  27.19  0.007510  0.03345  0.03672
## 6  0.1794  0.05742 0.4467  0.7732  3.180  53.91  0.004314  0.01382  0.02254
##   X0.01587 X0.03003 X0.006193 X25.38 X17.33 X184.6  X2019 X0.1622 X0.6656
## 1  0.01340  0.01389  0.003532  24.99  23.41 158.80 1956.0  0.1238  0.1866
## 2  0.02058  0.02250  0.004571  23.57  25.53 152.50 1709.0  0.1444  0.4245
## 3  0.01867  0.05963  0.009208  14.91  26.50  98.87  567.7  0.2098  0.8663
## 4  0.01885  0.01756  0.005115  22.54  16.67 152.20 1575.0  0.1374  0.2050
## 5  0.01137  0.02165  0.005082  15.47  23.75 103.40  741.6  0.1791  0.5249
## 6  0.01039  0.01369  0.002179  22.88  27.66 153.20 1606.0  0.1442  0.2576
##   X0.7119 X0.2654 X0.4601 X0.1189
## 1  0.2416  0.1860  0.2750 0.08902
## 2  0.4504  0.2430  0.3613 0.08758
## 3  0.6869  0.2575  0.6638 0.17300
## 4  0.4000  0.1625  0.2364 0.07678
## 5  0.5355  0.1741  0.3985 0.12440
## 6  0.3784  0.1932  0.3063 0.08368
```

```r
correct_column_names <- c(
  "ID", "Diagnosis", "radius_mean", "texture_mean",
  "perimeter_mean", "area_mean", "smoothness_mean", "compactness_mean",
  "concavity_mean", "concave_points_mean", "symmetry_mean", "fractal_dimension_mean",
  "radius_se", "texture_se", "perimeter_se", "area_se",
  "smoothness_se", "compactness_se", "concavity_se", "concave_points_se",
  "symmetry_se", "fractal_dimension_se", "radius_worst", "texture_worst",
  "perimeter_worst", "area_worst", "smoothness_worst", "compactness_worst",
  "concavity_worst", "concave_points_worst", "symmetry_worst", "fractal_dimension_worst"
)

colnames(wdbc) <- correct_column_names
head(wdbc)
```

```
##         ID Diagnosis radius_mean texture_mean perimeter_mean area_mean
## 1   842517         M       20.57        17.77         132.90    1326.0
## 2 84300903         M       19.69        21.25         130.00    1203.0
## 3 84348301         M       11.42        20.38          77.58     386.1
## 4 84358402         M       20.29        14.34         135.10    1297.0
## 5   843786         M       12.45        15.70          82.57     477.1
## 6   844359         M       18.25        19.98         119.60    1040.0
##   smoothness_mean compactness_mean concavity_mean concave_points_mean
## 1         0.08474          0.07864         0.0869             0.07017
## 2         0.10960          0.15990         0.1974             0.12790
## 3         0.14250          0.28390         0.2414             0.10520
## 4         0.10030          0.13280         0.1980             0.10430
## 5         0.12780          0.17000         0.1578             0.08089
## 6         0.09463          0.10900         0.1127             0.07400
##   symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
## 1        0.1812                0.05667    0.5435     0.7339        3.398
## 2        0.2069                0.05999    0.7456     0.7869        4.585
## 3        0.2597                0.09744    0.4956     1.1560        3.445
## 4        0.1809                0.05883    0.7572     0.7813        5.438
## 5        0.2087                0.07613    0.3345     0.8902        2.217
## 6        0.1794                0.05742    0.4467     0.7732        3.180
##   area_se smoothness_se compactness_se concavity_se concave_points_se
## 1   74.08      0.005225        0.01308      0.01860           0.01340
## 2   94.03      0.006150        0.04006      0.03832           0.02058
## 3   27.23      0.009110        0.07458      0.05661           0.01867
## 4   94.44      0.011490        0.02461      0.05688           0.01885
## 5   27.19      0.007510        0.03345      0.03672           0.01137
## 6   53.91      0.004314        0.01382      0.02254           0.01039
##   symmetry_se fractal_dimension_se radius_worst texture_worst perimeter_worst
## 1     0.01389             0.003532        24.99         23.41          158.80
## 2     0.02250             0.004571        23.57         25.53          152.50
## 3     0.05963             0.009208        14.91         26.50           98.87
## 4     0.01756             0.005115        22.54         16.67          152.20
## 5     0.02165             0.005082        15.47         23.75          103.40
## 6     0.01369             0.002179        22.88         27.66          153.20
##   area_worst smoothness_worst compactness_worst concavity_worst
## 1     1956.0           0.1238            0.1866          0.2416
## 2     1709.0           0.1444            0.4245          0.4504
## 3      567.7           0.2098            0.8663          0.6869
## 4     1575.0           0.1374            0.2050          0.4000
## 5      741.6           0.1791            0.5249          0.5355
## 6     1606.0           0.1442            0.2576          0.3784
##   concave_points_worst symmetry_worst fractal_dimension_worst
## 1               0.1860         0.2750                 0.08902
## 2               0.2430         0.3613                 0.08758
## 3               0.2575         0.6638                 0.17300
## 4               0.1625         0.2364                 0.07678
## 5               0.1741         0.3985                 0.12440
## 6               0.1932         0.3063                 0.08368
```

### 2.1.2. Checking Data Structure

```
str(wdbc)
```

```
## ’data.frame’:    568 obs. of  32 variables:
##  $ ID                     : int  842517 84300903 84348301 84358402 843786 844359 84458202 844981 8456
##  $ Diagnosis              : chr  "M" "M" "M" "M" ...
##  $ radius_mean            : num  20.6 19.7 11.4 20.3 12.4 ...
##  $ texture_mean           : num  17.8 21.2 20.4 14.3 15.7 ...
##  $ perimeter_mean         : num  132.9 130 77.6 135.1 82.6 ...
##  $ area_mean              : num  1326 1203 386 1297 477 ...
##  $ smoothness_mean        : num  0.0847 0.1096 0.1425 0.1003 0.1278 ...
##  $ compactness_mean       : num  0.0786 0.1599 0.2839 0.1328 0.17 ...
```

Not sure how to keep
these output within
page.

```
##  $ concavity_mean         : num  0.0869 0.1974 0.2414 0.198 0.1578 ...
##  $ concave_points_mean    : num  0.0702 0.1279 0.1052 0.1043 0.0809 ...
##  $ symmetry_mean          : num  0.181 0.207 0.26 0.181 0.209 ...
##  $ fractal_dimension_mean : num  0.0567 0.06 0.0974 0.0588 0.0761 ...
##  $ radius_se              : num  0.543 0.746 0.496 0.757 0.335 ...
##  $ texture_se             : num  0.734 0.787 1.156 0.781 0.89 ...
##  $ perimeter_se           : num  3.4 4.58 3.44 5.44 2.22 ...
##  $ area_se                : num  74.1 94 27.2 94.4 27.2 ...
##  $ smoothness_se          : num  0.00522 0.00615 0.00911 0.01149 0.00751 ...
##  $ compactness_se         : num  0.0131 0.0401 0.0746 0.0246 0.0335 ...
##  $ concavity_se           : num  0.0186 0.0383 0.0566 0.0569 0.0367 ...
##  $ concave_points_se      : num  0.0134 0.0206 0.0187 0.0188 0.0114 ...
##  $ symmetry_se            : num  0.0139 0.0225 0.0596 0.0176 0.0216 ...
##  $ fractal_dimension_se   : num  0.00353 0.00457 0.00921 0.00511 0.00508 ...
##  $ radius_worst           : num  25 23.6 14.9 22.5 15.5 ...
##  $ texture_worst          : num  23.4 25.5 26.5 16.7 23.8 ...
##  $ perimeter_worst        : num  158.8 152.5 98.9 152.2 103.4 ...
##  $ area_worst             : num  1956 1709 568 1575 742 ...
##  $ smoothness_worst       : num  0.124 0.144 0.21 0.137 0.179 ...
##  $ compactness_worst      : num  0.187 0.424 0.866 0.205 0.525 ...
##  $ concavity_worst        : num  0.242 0.45 0.687 0.4 0.535 ...
##  $ concave_points_worst   : num  0.186 0.243 0.258 0.163 0.174 ...
##  $ symmetry_worst         : num  0.275 0.361 0.664 0.236 0.399 ...
##  $ fractal_dimension_worst: num  0.089 0.0876 0.173 0.0768 0.1244 ...
```

We can see that our data is comprised of 568 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

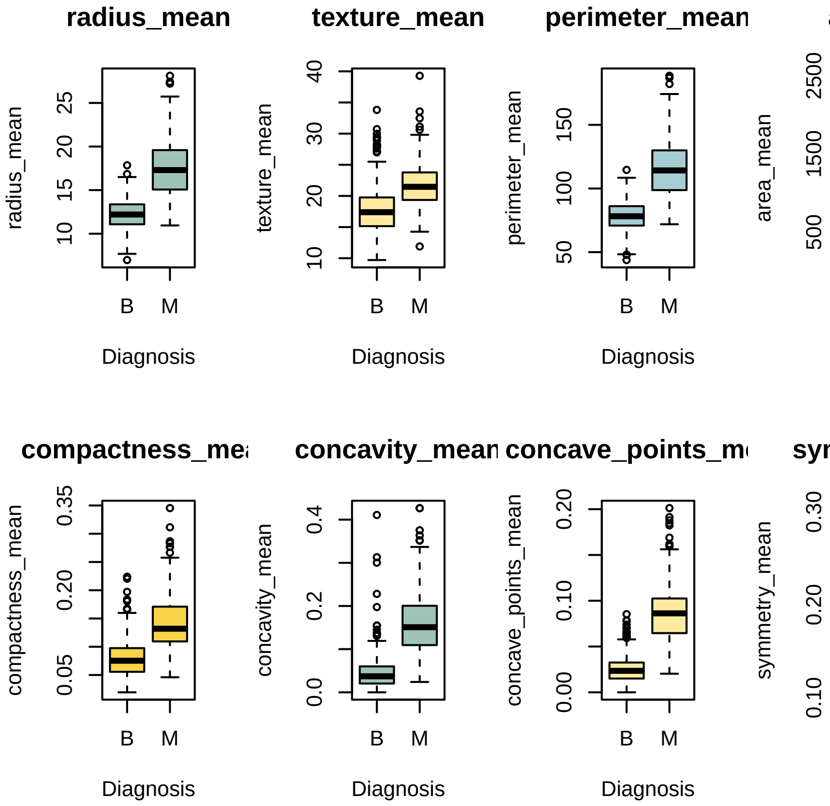
## 2.3. Exploratory Data Analysis

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

### 2.3.1. Checking Missing Values

```
##                      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
```

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

par(mfrow = c(2, 5))

for (i in 3:12) {
  column_name <- colnames(wdbc)[i]
  boxplot(wdbc[[column_name]] ~ wdbc$Diagnosis,
          xlab = "Diagnosis", ylab = column_name,
          main = paste(column_name),
          col = color_palette[i %% length(color_palette) + 1])
}
```



I don't know why it shows like this but can edit on pdf tools.

Or maybe just knit on word then export to pdf.

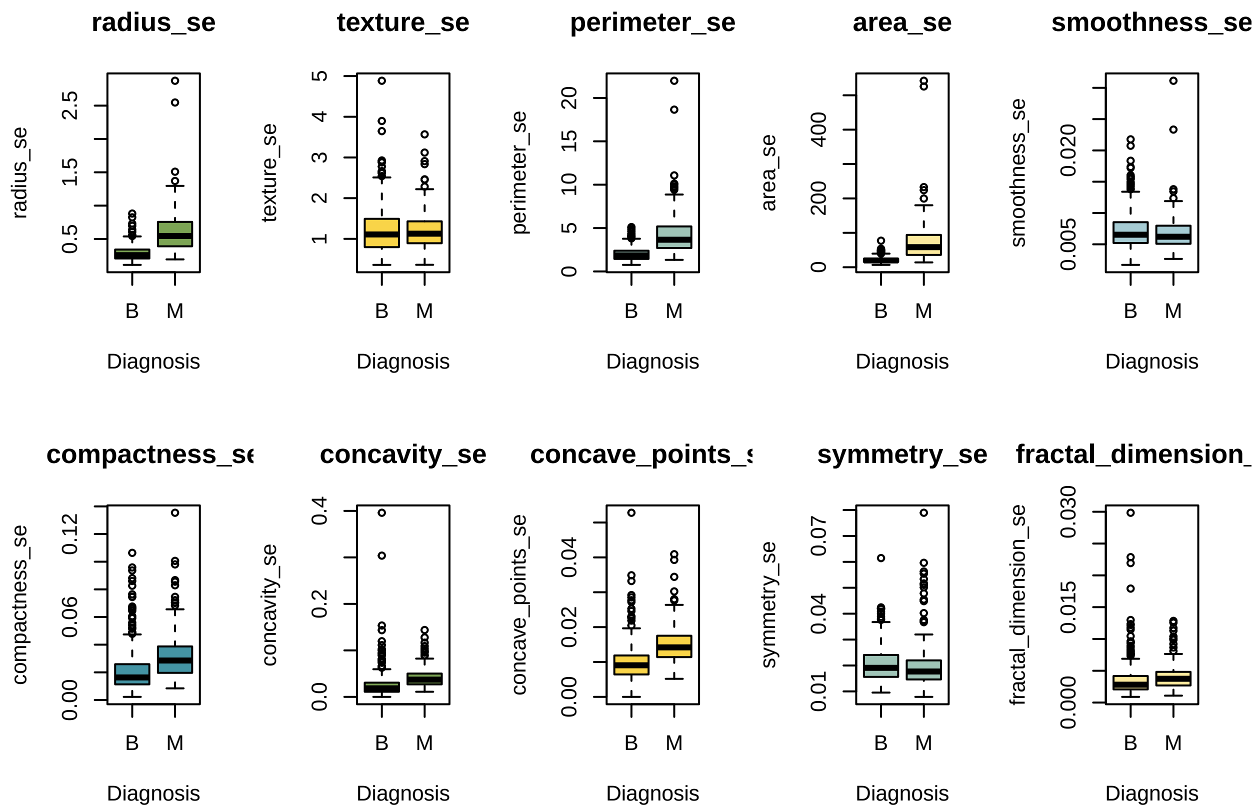
But the pdf knit is nicer.

### 2.3.2. Boxplots for Different Feature Groups

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

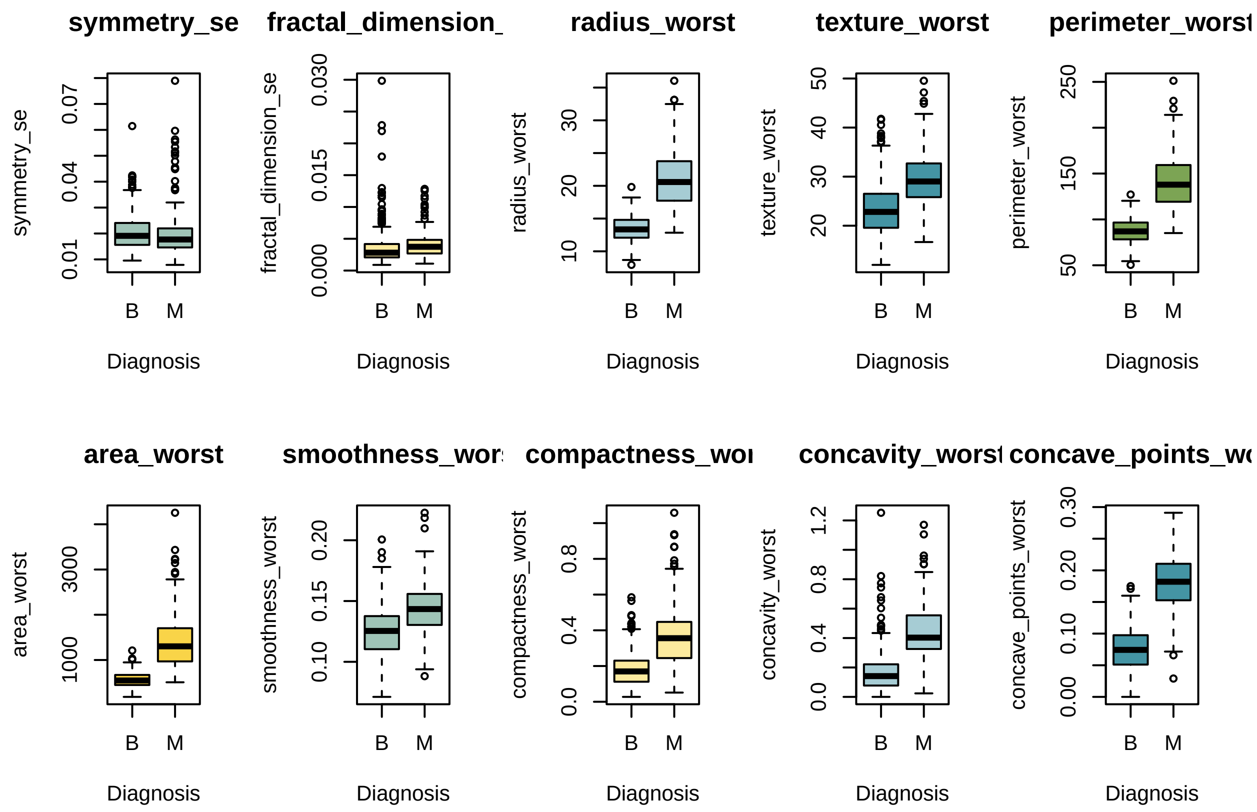
```
par(mfrow = c(2, 5))

for (i in 13:22) {
  column_name <- colnames(wdbc)[i]
  boxplot(wdbc[[column_name]] ~ wdbc$Diagnosis,
          xlab = "Diagnosis", ylab = column_name,
          main = paste(column_name),
          col = color_palette[i %% length(color_palette) + 1])
}
```

I revised this EDA part as it makes more sense to differentiate the two groups per variable and show that each group is distinct.



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

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

for (i in 21:30) {
  column_name <- colnames(wdbc)[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))
```

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.

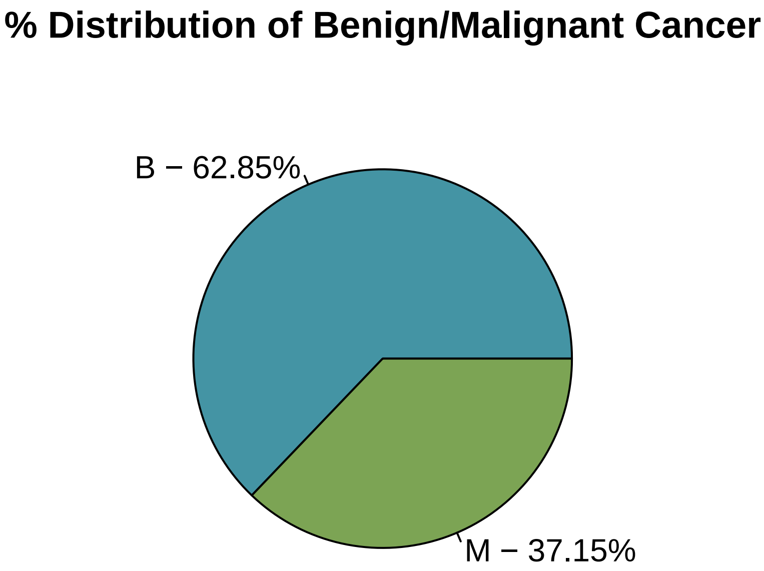
Several outliers are observed across the features; however, given the clinical nature of the data, the outliers have been retained, as they likely represent natural variations rather than measurement errors.

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

### 2.3.3. Pie Chart for Diagnosis Distribution

```
##
##        B        M
## 62.85211 37.14789
```

```r
pie(diagnosis_rel_freq,
    main = "% Distribution of Benign/Malignant Cancer",
    labels = c("B - 62.85%", "M - 37.15%"),
    col = color_palette)
```



Observed diagnosis is 63% Benign (B) and 37% Malignant (M).

### 2.4. Data Pre-processing

```r
diagnosis <- c(wdbc$Diagnosis)
wdbc_numerical <- wdbc[, -c(1, 2)]

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
## 842517     1.8304703   -0.3580115      1.6876185  1.9087096      -0.8245388
## 84300903   1.5807127    0.4534599      1.5682074  1.5592021       0.9453591
## 84348301  -0.7664412    0.2505920     -0.5902513 -0.7620394       3.2876616
## 84358402   1.7510020   -1.1578238      1.7782062  1.8263054       0.2832492
## 843786    -0.4741112   -0.8406971     -0.3847818 -0.5034606       2.2411009
## 844359     1.1720184    0.1573195      1.1399745  1.0960336      -0.1204242
##          compactness_mean concavity_mean concave_points_mean symmetry_mean
## 842517        -0.48506996    -0.01926127           0.5547716   0.005310158
## 84300903       1.06701941     1.37428289           2.0497027   0.946033074
## 84348301       3.43545507     1.92917831           1.4618812   2.878724511
## 84358402       0.54940161     1.38184965           1.4385755  -0.005671043
## 843786         1.25993231     0.87487702           0.8323684   1.011920282
## 844359         0.09481477     0.30610922           0.6539503  -0.060577050
##          fractal_dimension_mean  radius_se texture_se perimeter_se    area_se
## 842517               -0.8670525  0.5055180 -0.8759421    0.2697554  0.7495621
## 84300903             -0.3951172  1.2376623 -0.7799240    0.8605118  1.1901306
## 84348301              4.9283702  0.3319915 -0.1112396    0.2931467 -0.2850561
## 84358402             -0.5600103  1.2796855 -0.7900693    1.2850402  1.1991848
## 843786                1.8991708 -0.2516228 -0.5927793   -0.3180149 -0.2859394
## 844359               -0.7604406  0.1548423 -0.8047438    0.1612593  0.3041352
##          smoothness_se compactness_se concavity_se concave_points_se
## 842517      -0.6046865    -0.69044785   -0.4389349        0.26096614
## 84300903    -0.2968706     0.81710257    0.2140729        1.42403005
## 84348301     0.6881401     2.74596248    0.8197276        1.11463561
## 84358402     1.4801420    -0.04619076    0.8286684        1.14379320
## 843786       0.1557019     0.44775830    0.1610905       -0.06786669
## 844359      -0.9078435    -0.64909917   -0.3084658       -0.22661357
##          symmetry_se fractal_dimension_se radius_worst texture_worst
## 842517    -0.8029480          -0.09774547    1.8117525   -0.37154751
## 84300903   0.2389152           0.29485083    1.5172885   -0.02636404
## 84348301   4.7318744           2.04698660   -0.2785277    0.13157368
## 84358402  -0.3588553           0.50040655    1.3036983   -1.46897041
## 843786     0.1360599           0.48793718   -0.1624010   -0.31618789
## 844359    -0.8271492          -0.60898974    1.3742038    0.32044765
##          perimeter_worst area_worst smoothness_worst compactness_worst
## 842517         1.5437022  1.8973817       -0.3732153       -0.42767386
## 84300903       1.3555003  1.4624048        0.5295757        1.09222905
## 84348301      -0.2466058 -0.5474701        3.3957178        3.91481462
## 84358402       1.3465383  1.2264255        0.2228021       -0.31011937
## 843786        -0.1112797 -0.2412253        2.0502964        1.73366769
## 844359         1.3764116  1.2810177        0.5208107        0.02593314
##          concavity_worst concave_points_worst symmetry_worst
## 842517        -0.1433461            1.0942976     -0.2402319
## 84300903       0.8605552            1.9647360      1.1628329
## 84348301       1.9976368            2.1861633      6.0808757
## 84358402       0.6182342            0.7354327     -0.8677904
## 843786         1.2697122            0.9125746      1.7676302
## 844359         0.5143823            1.2042477      0.2686433
##          fractal_dimension_worst
## 842517                0.28504262
## 84300903              0.20511980
## 84348301              4.94609704
## 84358402             -0.39430135
## 843786                2.24870189
## 844359               -0.01133784
```

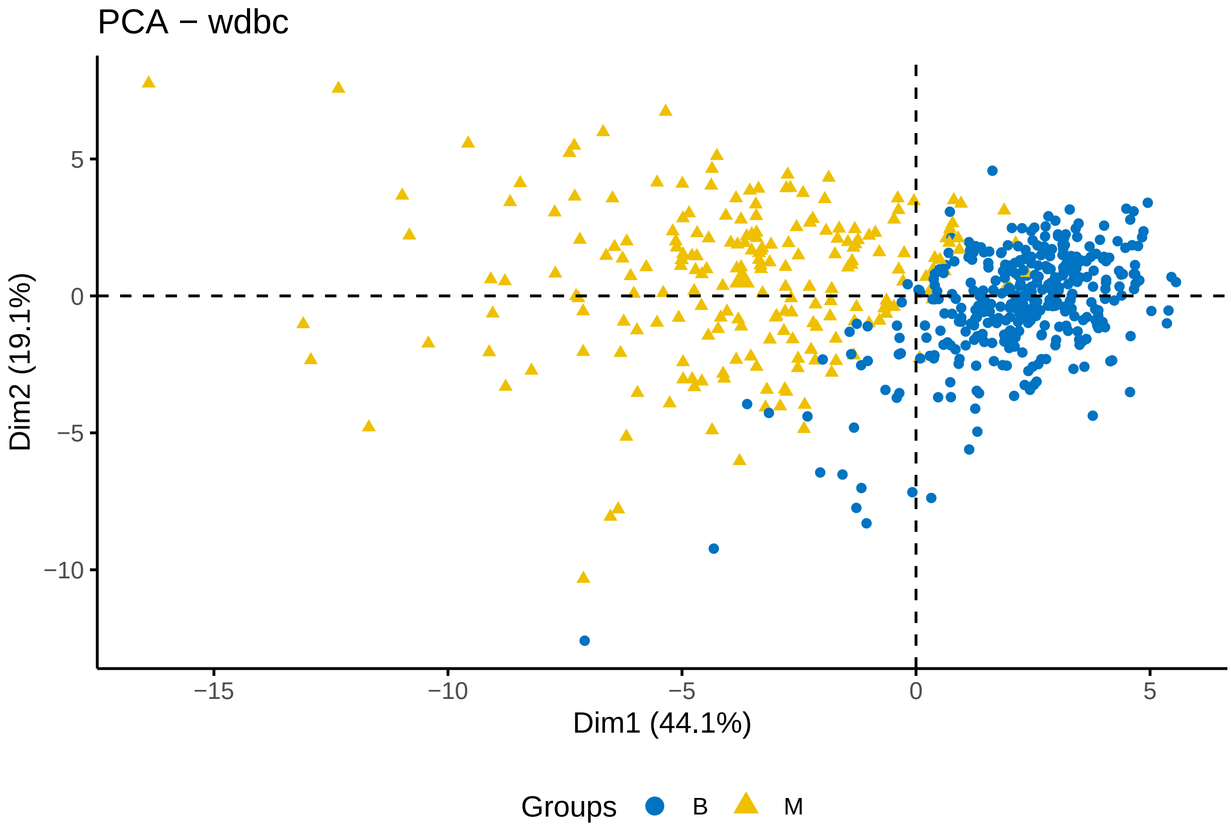
Our features have different scale of measurements so we standardized the data to ensure each variable contributes equally to the distance calculations, preventing variables with larger scales to have more weight in the clustering results.

# 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

```r
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.

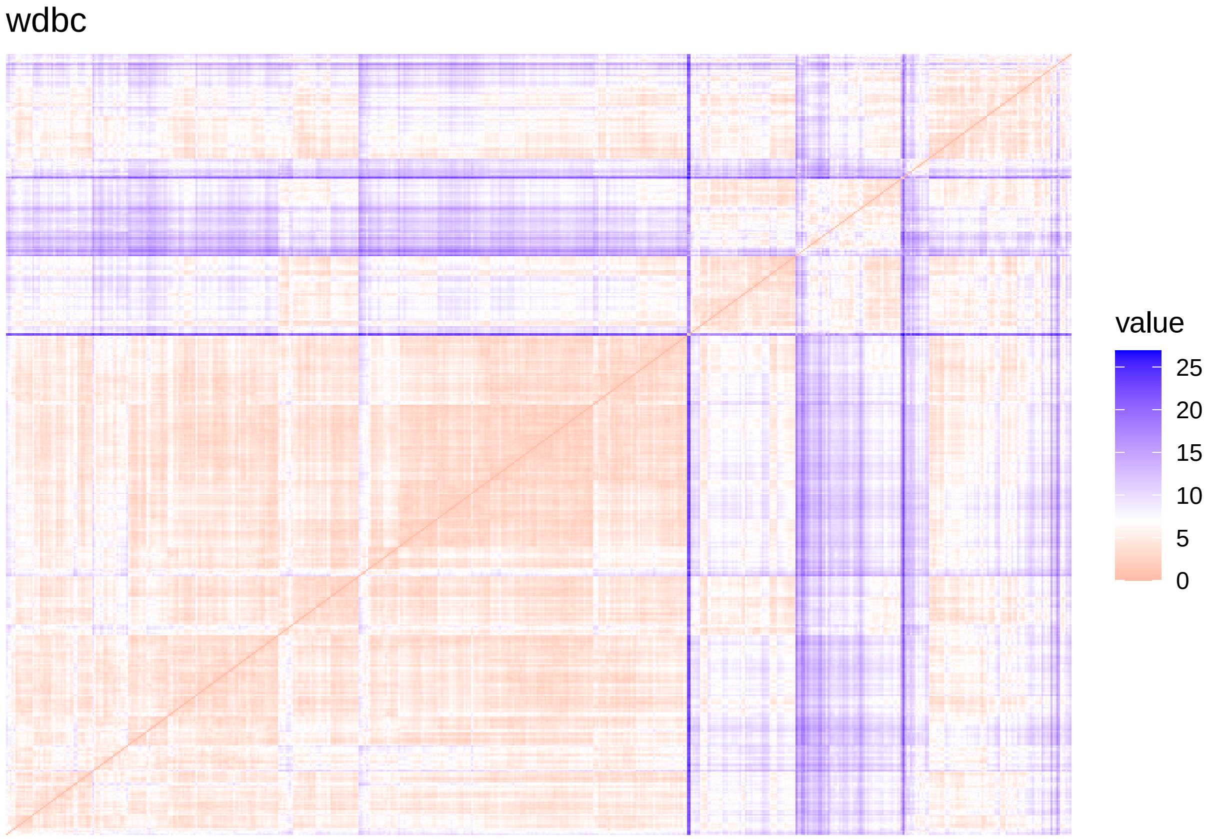
### 3.1.1. Hopkins Statistics

```r
set.seed(25)

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

```
## [1] 0.9999999
```

### 3.1.2. Visual Assessment of Cluster Tendency (VAT)

```r
fviz_dist(
  dist(wdbc),
  show_labels = FALSE
) + labs(title = "wdbc")
```

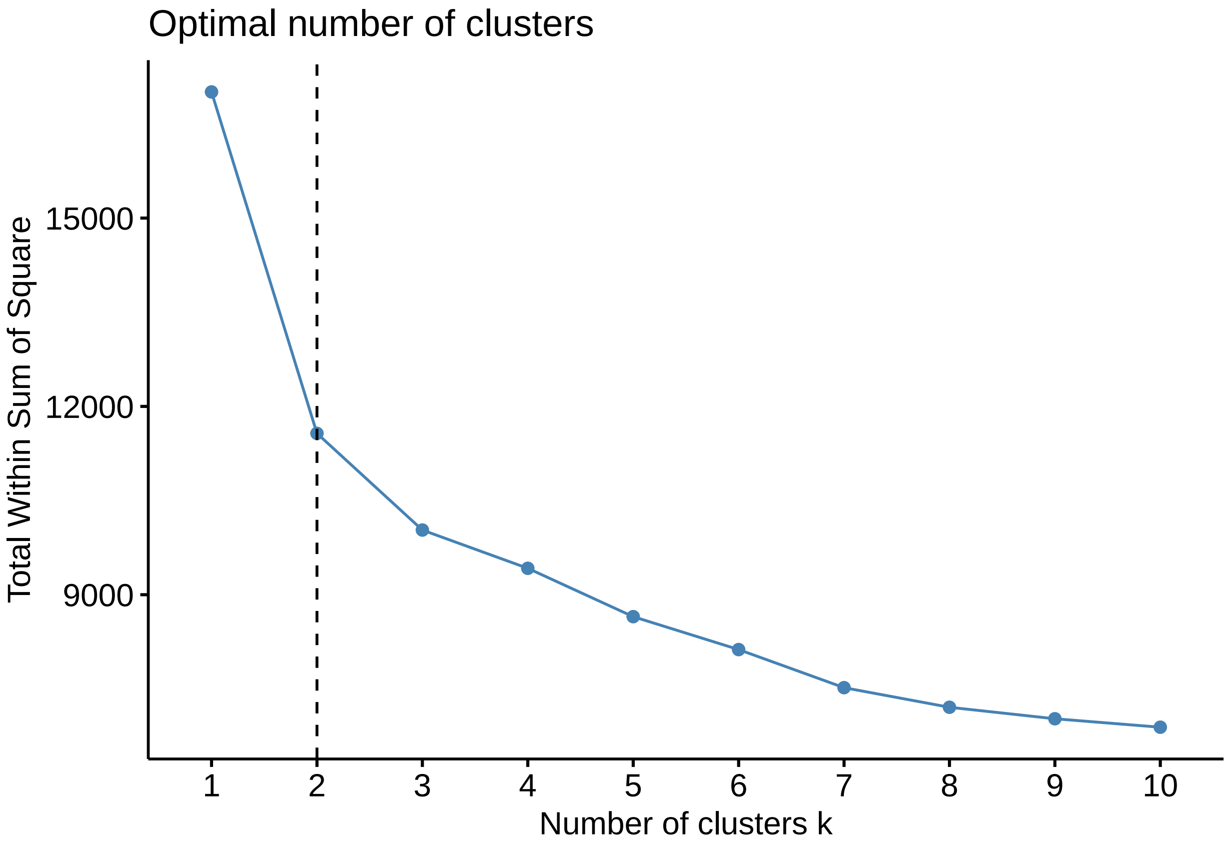


Based on the visual assessment and the Hopkins statistic of 0.9999999, the breast cancer dataset is confirmed to be suitable for clustering. Before proceeding with the partitioning clustering analysis, it is essential to determine the optimal number of clusters.

## 3.2. Finding the Optimal Number of Clusters

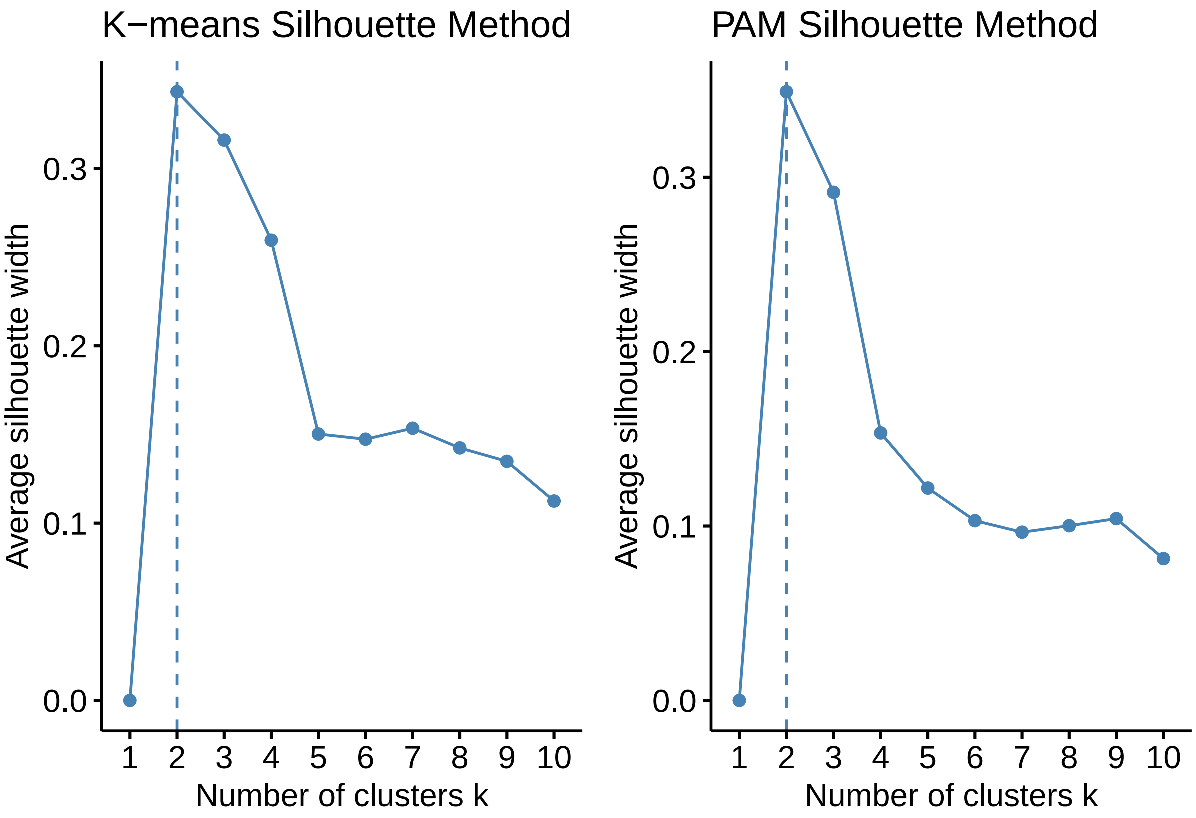
### 3.2.1. Elbow Method

```
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:

```
calculate_gap_kmeans <- function(data, max_clusters = 10, B = 10) {
  gap_stat <- clusGap(data, FUN = kmeans, K.max = max_clusters, B = B)
  gap_stat_values <- as.data.frame(gap_stat$Tab)
  return(gap_stat_values)
}
```

```
set.seed(101)
wdbc_matrix <- as.matrix(wdbc)
max_clusters <- 10

gap_stat_values <- calculate_gap_kmeans(wdbc_matrix, max_clusters, B = 10)
gap_values <- gap_stat_values$gap
gap_diff <- gap_values[-length(gap_values)] - gap_values[-1] - gap_stat_values$SE.sim[-1]

## Creating data frame for plotting
gap_stat_data <- data.frame(
  Clusters = 1:(max_clusters - 1),
  Gap_Diff = gap_diff
)

## Creating the bar plot
```

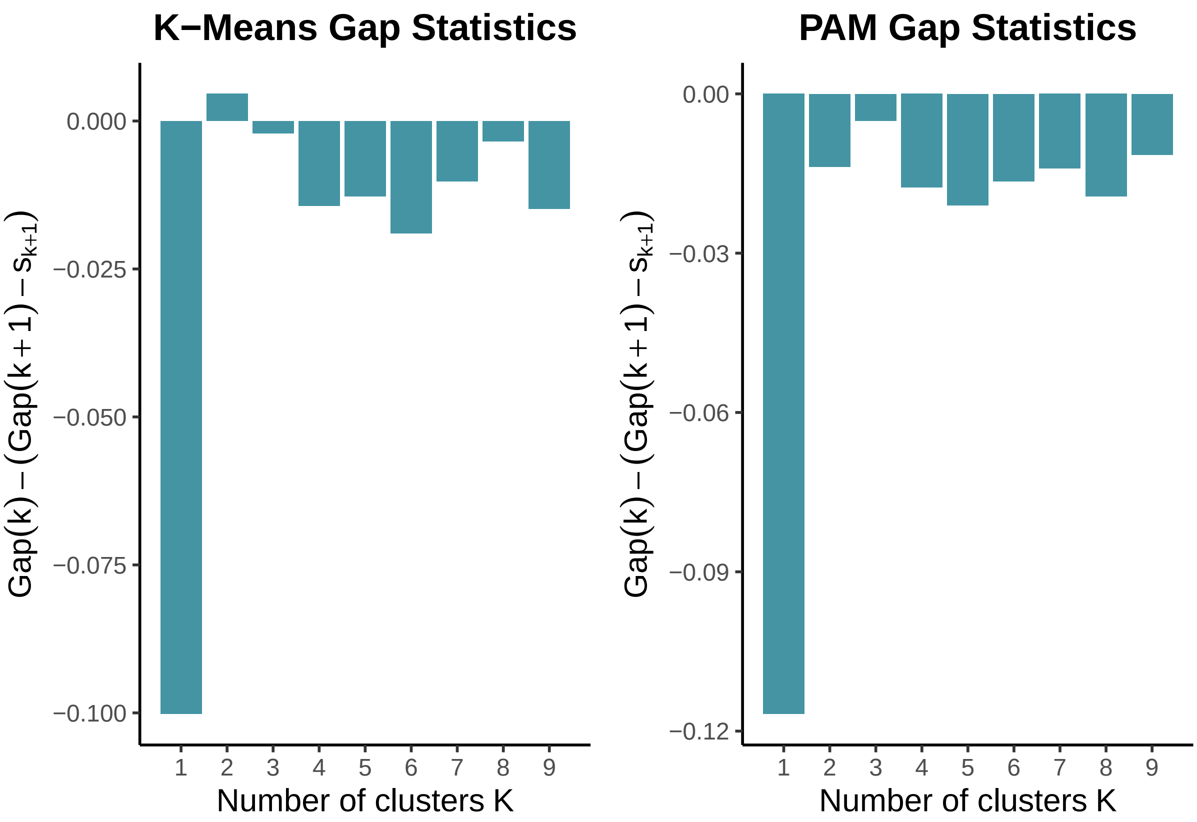
```
kmeans_gap <- ggplot(gap_stat_data, aes(x = Clusters, y = Gap_Diff)) +
  geom_bar(stat = "identity", fill = "#4494a4") +
  labs(title = "K-Means Gap Statistics",
       x = expression("Number of clusters"~K),
       y = expression(Gap(k) - (Gap(k + 1) - s[k + 1]))) +
  scale_x_continuous(breaks = 1:max_clusters) +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12)
  )
```

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

```
calculate_gap_pam <- function(data, max_clusters = 10, B = 10) {
  gap_stat <- clusGap(data, FUN = pam, K.max = max_clusters, B = B)
  gap_stat_values <- as.data.frame(gap_stat$Tab)
  return(gap_stat_values)
}
```

```
set.seed(101)
wdbc_matrix <- as.matrix(wdbc)
max_clusters <- 10

gap_stat_values <- calculate_gap_pam(wdbc_matrix, max_clusters, B = 10)
gap_values <- gap_stat_values$gap
gap_diff <- gap_values[-length(gap_values)] - gap_values[-1] - gap_stat_values$SE.sim[-1]

## Creating data frame for plotting
gap_stat_data <- data.frame(
  Clusters = 1:(max_clusters - 1),
  Gap_Diff = gap_diff
)

## Creating the bar plot
pam_gap <-ggplot(gap_stat_data, aes(x = Clusters, y = Gap_Diff)) +
  geom_bar(stat = "identity", fill = "#4494a4") +
  labs(title = "PAM Gap Statistics",
       x = expression("Number of clusters"~K),
       y = expression(Gap(k) - (Gap(k + 1) - s[k + 1]))) +
  scale_x_continuous(breaks = 1:max_clusters) +
  theme_classic() +
  theme(
    plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),
    axis.title = element_text(size = 12)
  )
```

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



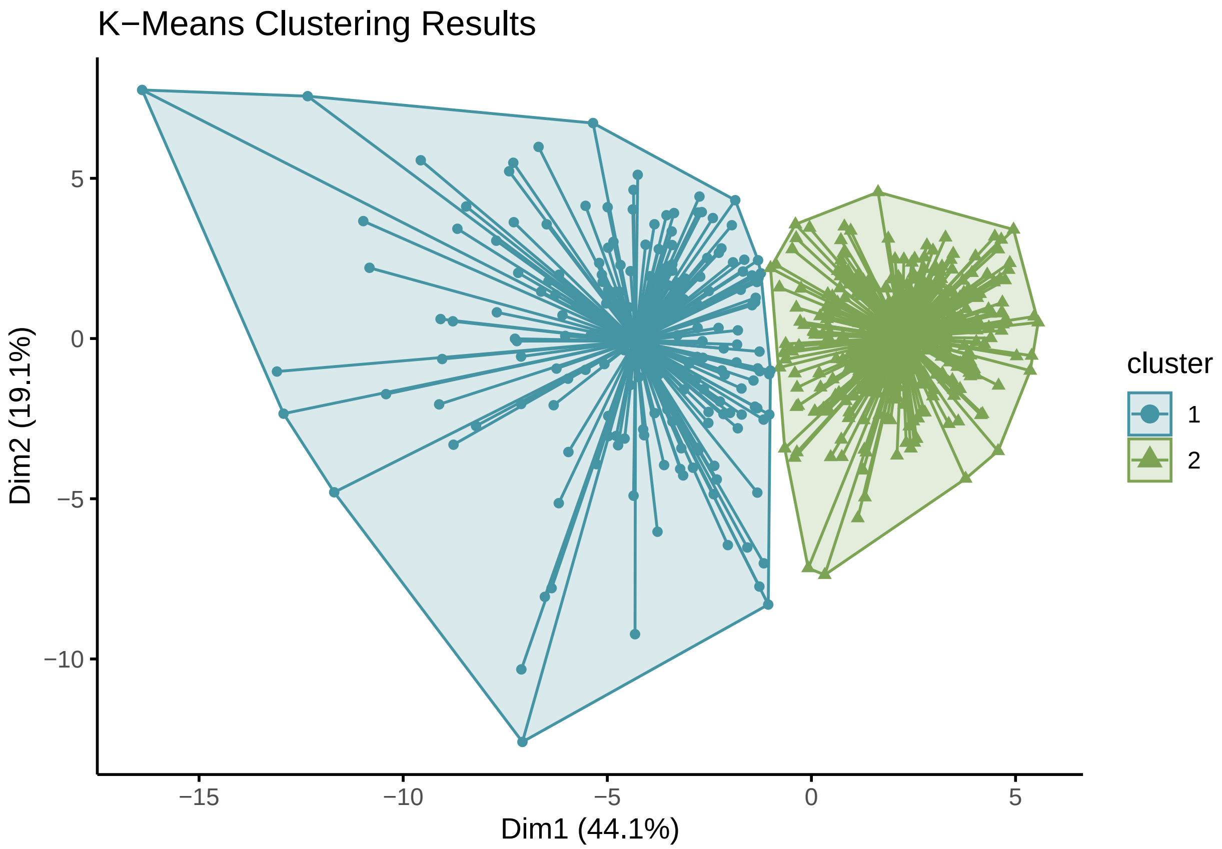
Both graphs show a notable drop in the gap statistic from 1 to 2 clusters, indicating a significant change in data structure. For the K-Means method, the gap statistic reaches its peak at K=2, with a positive value, before declining for higher values of K. In contrast, the PAM method shows negative gap statistics across all values of K, suggesting that no configuration from K=2 to K=9 offers a better clustering solution than random clustering. However, the most pronounced change still occurs between K=1 and K=2, reinforcing that two clusters best represent the data.

Overall, the elbow, silhouette, and gap statistics methods all suggest K=2 as the optimal number of clusters.

## 4. Clustering Analysis

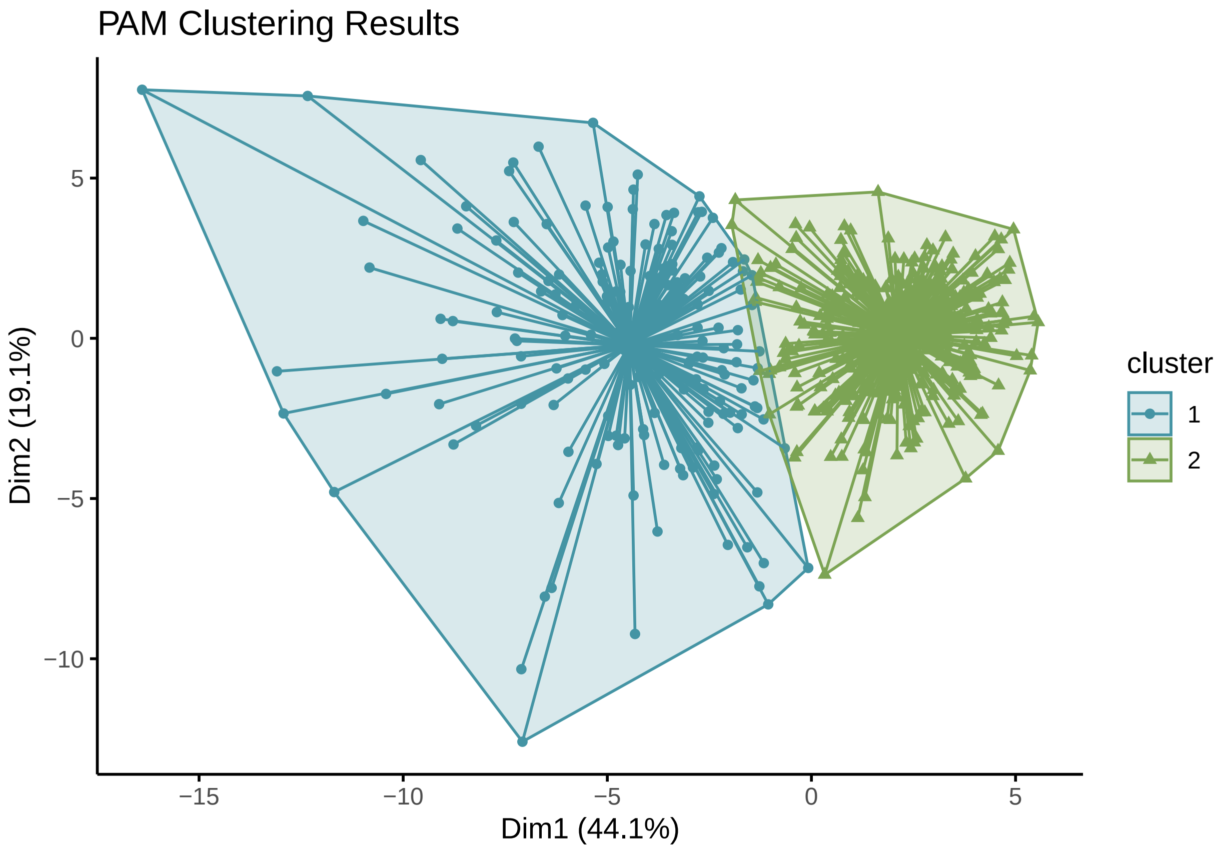
### 4.1. K-Means Clustering

```
set.seed(101)

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

kmeans_graph <- fviz_cluster(
  km.res,
  data = wdbc,
  palette = color_palette,
  ellipse.type = "convex",
  star.plot = TRUE,
  ellipse = TRUE,
  geom = "point",
  main = "K-Means Clustering Results",
  ggtheme = theme_classic()
)
kmeans_graph
```



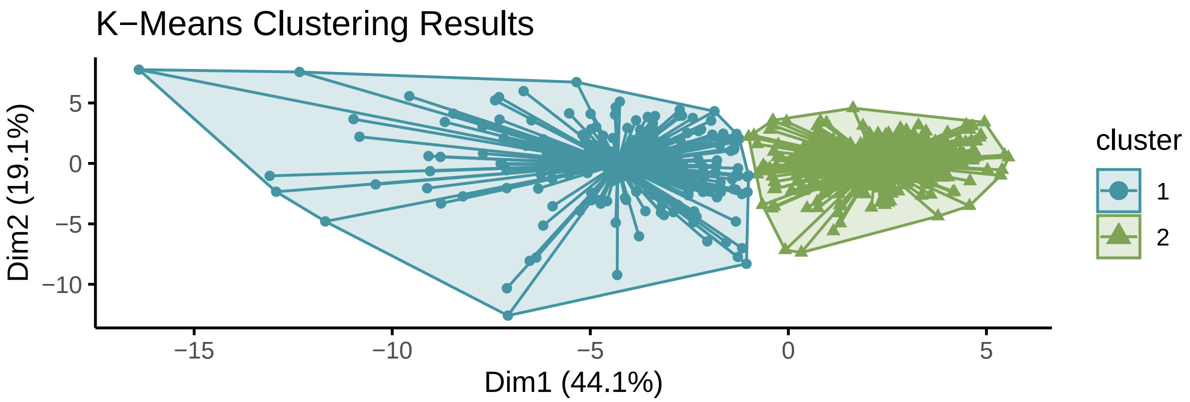
### 4.2. Partitional Around Medoid (PAM) Clustering

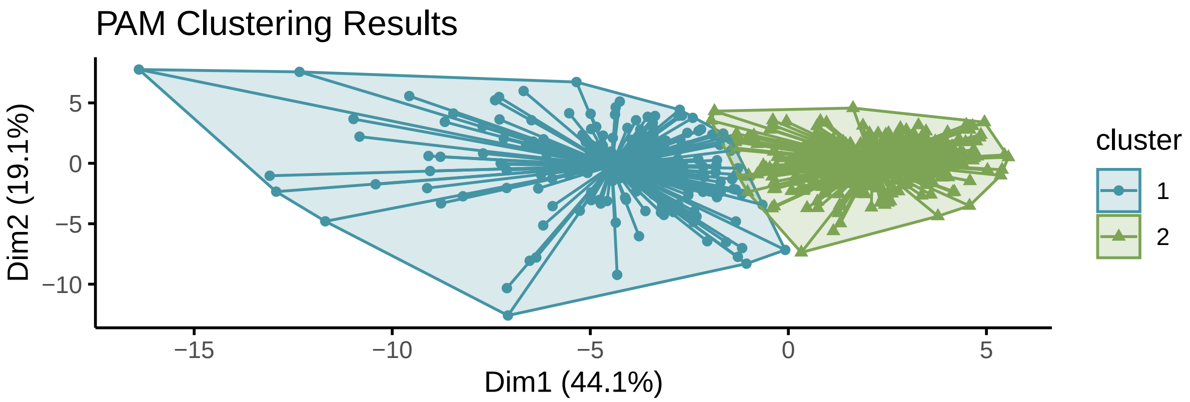
```
set.seed(101)

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

pam_graph <- fviz_cluster(
  pam.res,
  data = wdbc,
  palette = color_palette,
  ellipse.type = "convex",
  star.plot = TRUE,
  ellipse = TRUE,
  geom = "point",
  main = "PAM Clustering Results",
  ggtheme = theme_classic()
)
pam_graph
```



### 4.3. Comparing K-Means and PAM

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





I added an interpretation on this comparison. Not on this knitted file.

# 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 = km.res$cluster,
                      pam_cluster = pam.res$clustering)
```

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

```
##
##        1   2
##    1 175  36
##    2  18 339
```

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

```
##
##       1   2
##   1 166  45
##   2  17 340
```

The contingency table shows that the K-Means method has a 9.33% misclassification rate compared to the ground truth variable, representing the actual diagnosis. In comparison, the PAM method has a slightly higher misclassification rate of 10.92%.

### 5.1.2. Rand Index

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

```
##    Rand adjRand  Eindex
##  0.8276  0.6530  0.5850
```

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

```
##    Rand adjRand  Eindex
##  0.8052  0.6072  0.5976
```

The Rand Index (RI) for K-Means clustering is 0.8276, indicating a strong alignment between the clustering results and the actual diagnosis. For the PAM method, the Rand Index is slightly lower at 0.8052, but still reflects a good agreement with the actual diagnosis, though not as high as with K-Means.

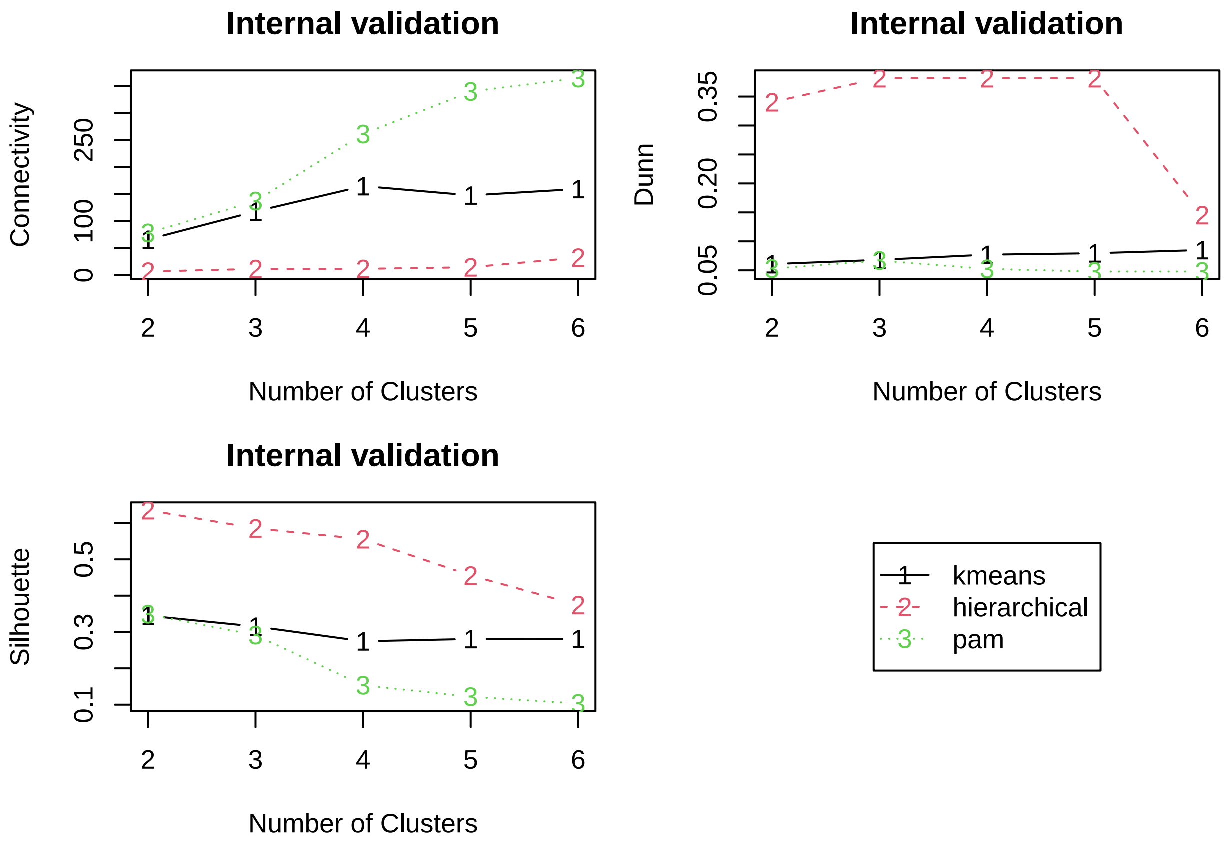
## 5.2. Internal Validation

```
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   66.0762 118.0032 164.9048 147.6623 159.5675
##              Dunn            0.0608   0.0681   0.0771   0.0795   0.0851
##              Silhouette      0.3449   0.3153   0.2745   0.2808   0.2811
## hierarchical Connectivity    6.7313  11.5893  11.7560  14.6849  32.6163
##              Dunn            0.3403   0.3818   0.3818   0.3818   0.1454
##              Silhouette      0.6356   0.5854   0.5556   0.4556   0.3743
## pam          Connectivity   78.0540 136.9873 261.0448 339.8480 364.6349
##              Dunn            0.0528   0.0668   0.0524   0.0479   0.0479
##              Silhouette      0.3490   0.2914   0.1534   0.1218   0.1031
##
## Optimal Scores:
##
##              Score  Method       Clusters
## Connectivity 6.7313 hierarchical 2
## Dunn         0.3818 hierarchical 3
## Silhouette   0.6356 hierarchical 2
```

```
op <- par(no.readonly = TRUE)
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))
```



```
par(op)
```

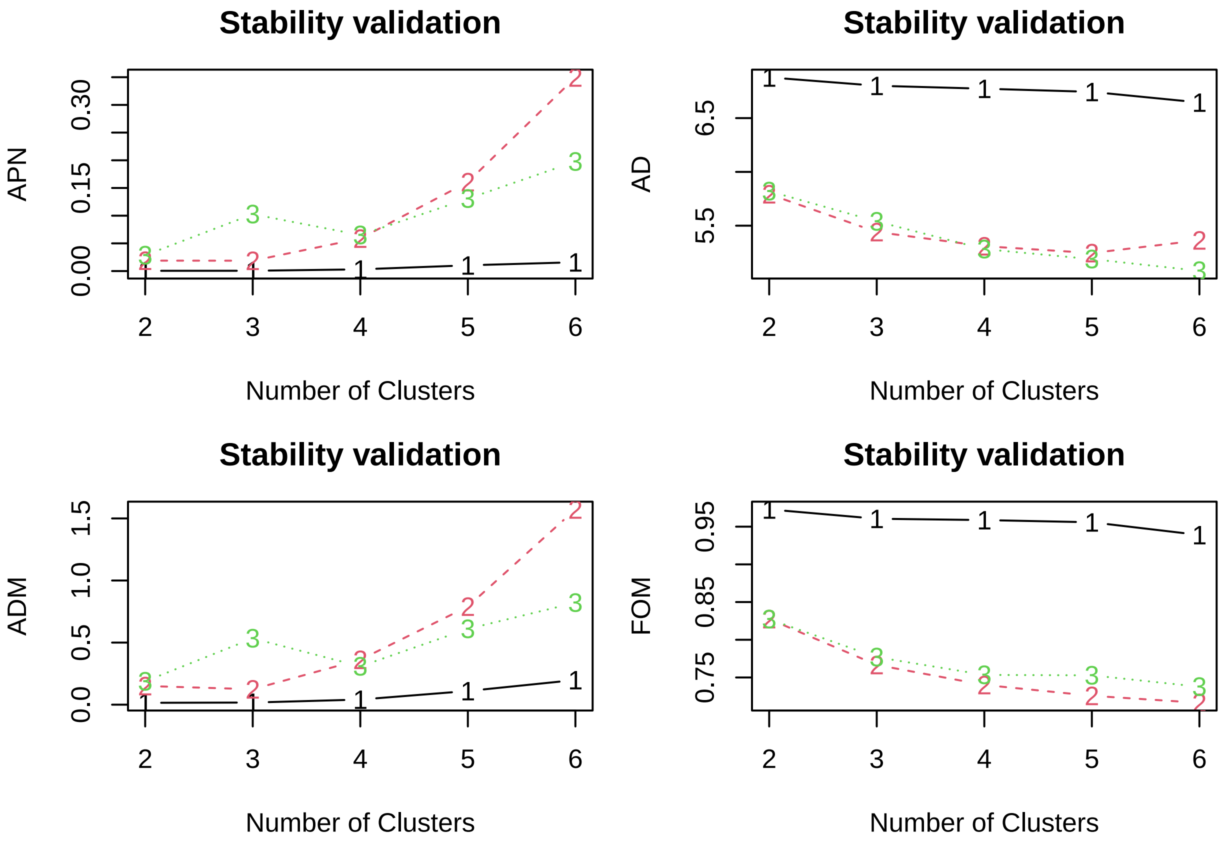
Hierarchical clustering shows the best performance based on the Dunn Index and Silhouette Score, suggesting well-defined and separated clusters. K-Means also performs relatively well but shows a decline in cluster quality as the number of clusters increases. PAM, while providing some separation, consistently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

### 5.3. Stability Validation

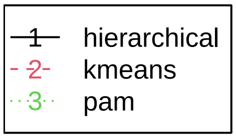
```
stab_wdbc <- clValid(wdbc,
                     nClust = 2:6,
                     clMethods = c("hierarchical", "kmeans", "pam"),
                     validation = "stability")

optimal_scores_stab <- optimalScores(stab_wdbc)
optimal_scores_stab
```

```
##             Score       Method Clusters
## APN 0.0004694836 hierarchical        2
## AD  5.0806805937          pam        6
## ADM 0.0149768081 hierarchical        2
## FOM 0.7164043908       kmeans        6
```

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

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



```r
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))
```



How to include the legends on the graphs?

**TO DO: Interpretation of stability**

From chatGPT - I don't want to take something I don't understand - Overall, hierarchical clustering shows the highest stability in terms of APN and ADM, especially with 2 clusters. The PAM method, while having the lowest AD score, indicates stable centroids but not necessarily the best clustering quality. K-Means provides the best clustering quality (lowest FOM) but is less stable than hierarchical clustering according to APN and ADM metrics.

## 6. Conclusion and Recommendation

In this study, we utilized partitional 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, as indicated by a lower misclassification rate and higher Rand Index.

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.

The following are recommended for further study:

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.