Factorial Analysis of Mixed Data + DBSCAN

Seyifunmi M. Owoeye

```r
set.seed(42)
```

Background

Factorial analysis of mixed data (FAMD) is a dimension-reduction technique that reduces the dimensionality of large data sets containing categorical and numerical features. It also aids in examining the relationship between all features (Pagès, 2004).

References

- Pagès, J. 2004. “Analyse Factorielle de Donnees Mixtes.” *Revue Statistique Appliquee* 4: 93–111.

```r
suppressPackageStartupMessages({
  library("FactoMineR")
  library("factoextra")
  library("corrplot")
  library(dbscan)
  library(fpc)
  library(FNN)
  library(flexmix)
  library(ggplot2)
  library(gridExtra)
})
```

```
Warning: package 'FactoMineR' was built under R version 4.3.2
```

```
Warning: package 'ggplot2' was built under R version 4.3.3
```

```
Warning: package 'dbscan' was built under R version 4.3.3

Warning: package 'fpc' was built under R version 4.3.3

Warning: package 'FNN' was built under R version 4.3.3

Warning: package 'flexmix' was built under R version 4.3.3
```

Load dataset

```
data <- read.csv("heart_disease.csv", header = T)
data <- subset(data, select = -c(disease))
```

```
res.famd <- FAMD(data, ncp = 13, sup.var = NULL, ind.sup = NULL, graph = FALSE)
res.famd
```

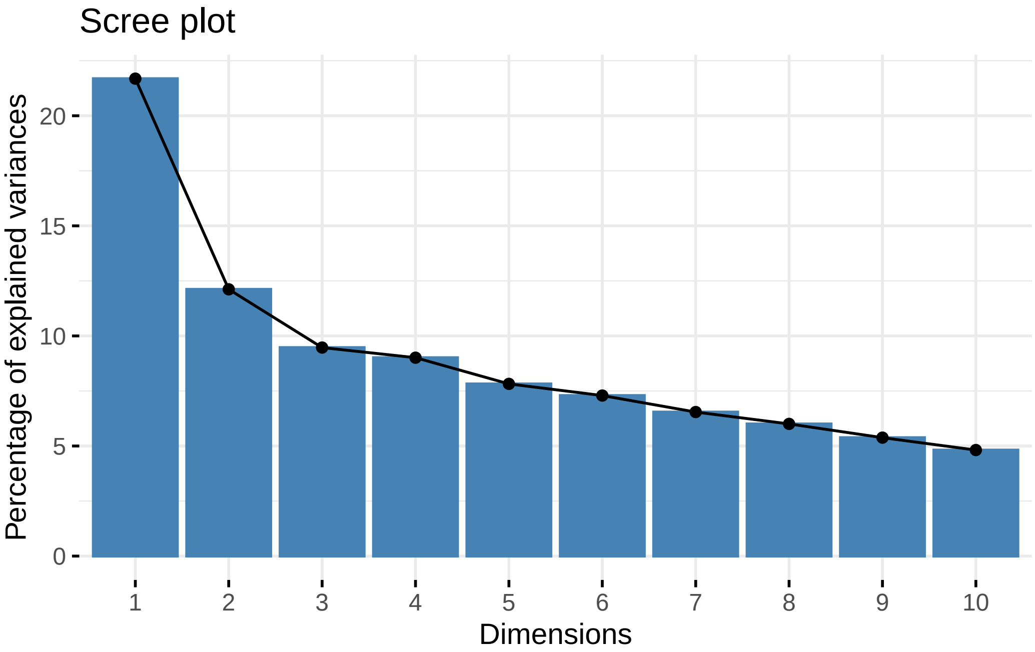
```
*The results are available in the following objects:

  name          description
1 "$eig"        "eigenvalues and inertia"
2 "$var"        "Results for the variables"
3 "$ind"        "results for the individuals"
4 "$quali.var"  "Results for the qualitative variables"
5 "$quanti.var" "Results for the quantitative variables"
```

The proportion of variances retained by the different dimensions are:

```
eig.val <- get_eigenvalue(res.famd)
print(head(eig.val,9))
```

```
      eigenvalue variance.percent cumulative.variance.percent
Dim.1  2.8190685        21.685142                    21.68514
Dim.2  1.5751568        12.116591                    33.80173
Dim.3  1.2311895         9.470689                    43.27242
Dim.4  1.1714378         9.011060                    52.28348
Dim.5  1.0167427         7.821097                    60.10458
Dim.6  0.9478786         7.291374                    67.39595
Dim.7  0.8504764         6.542126                    73.93808
Dim.8  0.7804035         6.003104                    79.94118
Dim.9  0.6992666         5.378974                    85.32016
```

```r
fviz_screeplot(res.famd)
```



As seen in the results above, we need the `first 8` dimensions to explain at least `80%` of the variability in the dataset.

The FAMD results for each data point are stored in the code chunk below. The coordinates from each dimension will be used to perform the clustering analysis.

```r
ind <- get_famd_ind(res.famd)
ind
```

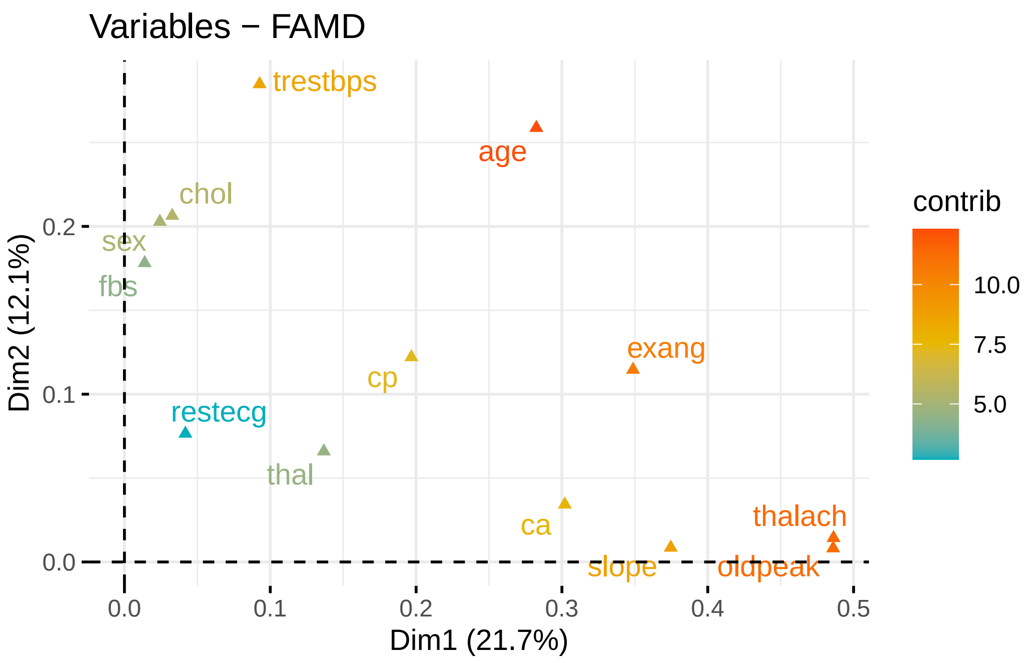
```
FAMD results for individuals
 ===================================================
  Name       Description                      
1 "$coord"   "Coordinates"                    
2 "$cos2"    "Cos2, quality of representation"
3 "$contrib" "Contributions"                  
```

```r
head(ind$coord)
```

```
       Dim.1         Dim.2       Dim.3      Dim.4      Dim.5      Dim.6
1  0.5155972  2.2417148229  3.54021942 -1.2534038  0.1567056 -1.7447860
2  3.5477530  0.9260241258 -0.75162025  0.5180071 -0.7460769 -0.9134170
3  3.0597387 -0.9186633420 -0.28590041  0.7244140 -0.2958555 -0.4477828
4 -0.5544128 -1.0807719094  2.24504247 -1.6361527  2.1105380  0.5215210
5 -1.8792096  0.0006278454 -0.15412516 -0.4735190  0.8975480 -0.8675078
6 -1.7452721 -0.4673007344 -0.02251221  0.1424091 -0.4002311  0.4358229
       Dim.7         Dim.8      Dim.9     Dim.10     Dim.11     Dim.12
1 -0.2533115  0.7242901763 -0.6765588  0.9599646 -0.5730676  0.6799776
2  0.2051667 -1.1041195315  0.5668879  1.1169526  0.4697796 -0.7565438
3  0.9201399 -0.1616525926 -0.1458584 -0.4876801  0.6619885  0.7853959
4  0.7317173  1.4290588122  1.2403156  1.2119925 -0.3430609  0.3478021
5 -0.1148671  0.0001802223  1.2402343 -1.0830559  0.5560790 -0.3035951
6  0.3612177 -0.2378157111  0.1097934  0.7434550 -0.4091675  1.0396145
      Dim.13
1 -0.1326154
2 -0.7610028
3  0.1770617
4 -0.2378025
5  1.0280130
6  0.4555453
```

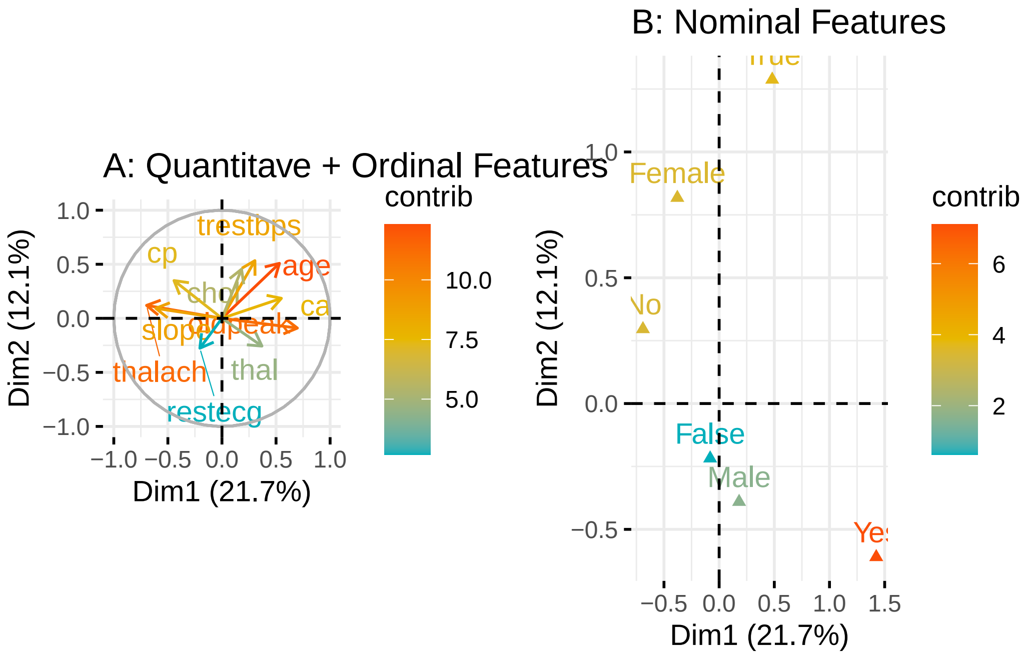
Accessing the correlation between features and their contributions to each component.

```r
fviz_famd_var(res.famd, repel = TRUE, col.var = "contrib",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"))
```



First 2 Dimensions or Components

```
# --Plot of variables
# fviz_famd_var(res.famd, repel = TRUE, col.var = "cos2", gradient.cols = c("#00AFBB", "#E

# # Contribution to the first dimension
# fviz_contrib(res.famd, "var", axes = 1:6)
# # Contribution to the second dimension
# fviz_contrib(res.famd, "var", axes = 2)

# Most contributing quantitative and ordianal features
p <- fviz_famd_var(res.famd, "quanti.var", col.var = "contrib",
                   gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
                   repel = TRUE)  +
          ggtitle("A: Quantitave + Ordinal Features")

q <- fviz_famd_var(res.famd, "quali.var", col.var = "contrib",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07")) +
                ggtitle("B: Nominal Features")

grid.arrange(p, q, ncol = 2)
```

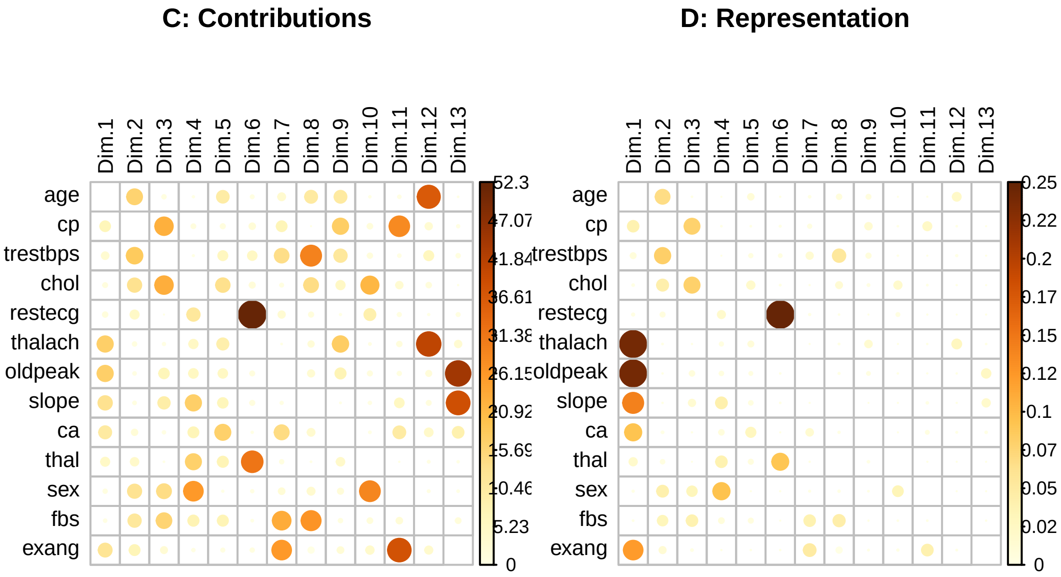


The figure above shows the relationship between the features, their percentage contribution, and the quality of their representation on the factor map. `Figure A` shows that `chol`, `trestbps`, `age` and `ca` are positively correlated and are negatively correlated with `slope`, `thalach` and `slope`. The idea is that positively correlated features are grouped while negatively correlated features are on a different quadrant in the plot. The figure also shows that `age`, `thalach`, `oldpeak`, `slope` and `trestbps` contribute most to the first and second dimensions. Additionally, the distance between the origin and each variable measures the quality of the variable representation on the map, with the most represented features being far away from the origin.

Of the three nominal features, `exang = Yes` contributed most to the first and second dimensions (`Figure B`).

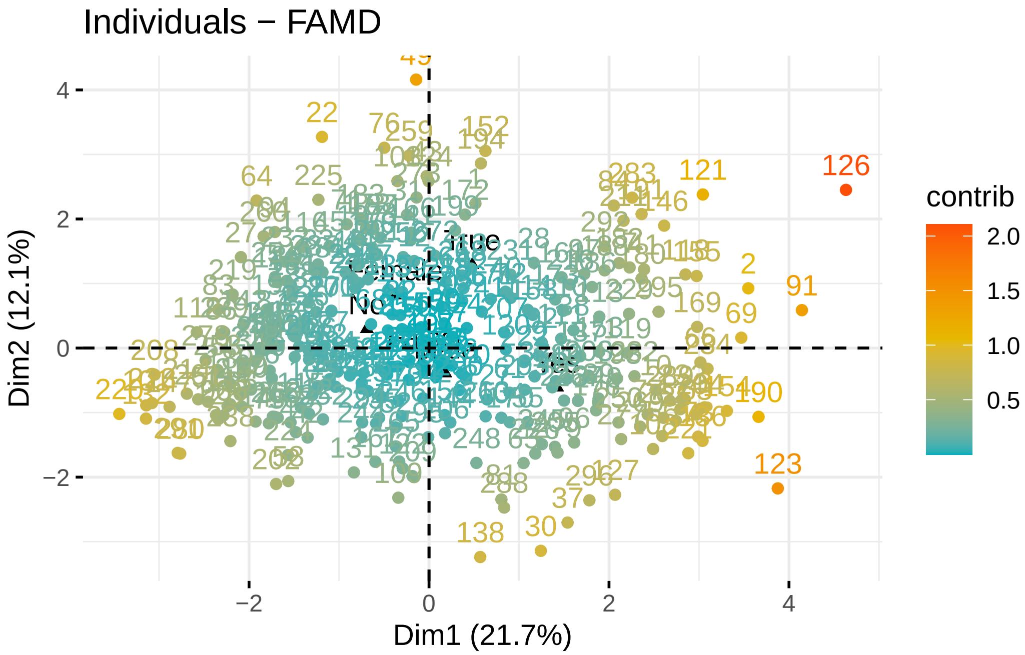
All Dimensions

```r
var <- get_famd_var(res.famd)

# Set up a 1x2 layout for plots
par(mfrow = c(1, 2), cex = 0.7)

corrplot(var$contrib, is.corr = FALSE, method = "circle",
         diag = FALSE, tl.col = "black",
         title = "C: Contributions", mar = c(1, 0, 1, 0))
corrplot(var$cos2, is.corr = FALSE, method = "circle",
         diag = FALSE, tl.col = "black",
         title = "D: Representation", mar = c(1, 0, 1, 0))
```



Individual Data Points

```
fviz_famd_ind(res.famd, col.ind = "contrib",
              gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = FALSE#
)
```



Conclusion: Based on the findings and observations above, the first 9 components will be used for clustering analysis. These components account for 80% of the variability in the data.

Extracting the Low-Dimension Data

```
ind <- get_famd_ind(res.famd)

famd_data <- as.data.frame(ind$coord[,1:8])
head(famd_data)
```

```
       Dim.1         Dim.2       Dim.3      Dim.4      Dim.5      Dim.6
1  0.5155972  2.2417148229  3.54021942 -1.2534038  0.1567056 -1.7447860
2  3.5477530  0.9260241258 -0.75162025  0.5180071 -0.7460769 -0.9134170
3  3.0597387 -0.9186633420 -0.28590041  0.7244140 -0.2958555 -0.4477828
4 -0.5544128 -1.0807719094  2.24504247 -1.6361527  2.1105380  0.5215210
5 -1.8792096  0.0006278454 -0.15412516 -0.4735190  0.8975480 -0.8675078
6 -1.7452721 -0.4673007344 -0.02251221  0.1424091 -0.4002311  0.4358229
       Dim.7         Dim.8
1 -0.2533115  0.7242901763
2  0.2051667 -1.1041195315
```

```
3  0.9201399 -0.1616525926
4  0.7317173  1.4290588122
5 -0.1148671  0.0001802223
6  0.3612177 -0.2378157111
```

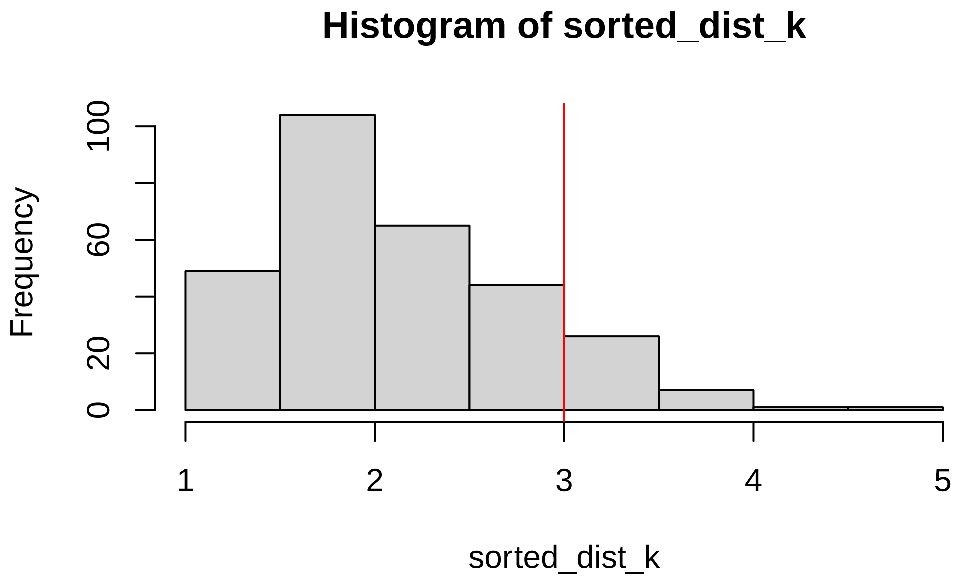
```
write.csv(famd_data, "heart_disease_reduced.csv")
```

Performing Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

Before employing DBSCAN, it is important to determine the optimal and `minimum points`. The `k-distance graph` would be used to estimate .

Estimating Using k-Distance Graph

```
k <- 5

dist_k <- knn.dist(famd_data, k = k)
sorted_dist_k <- sort(dist_k[, k])


hist(sorted_dist_k)
abline(v = 3, col ="red")
```



Based on the distribution of distances, our choice of `is 3` and the `minimum point is [4,5]`
.

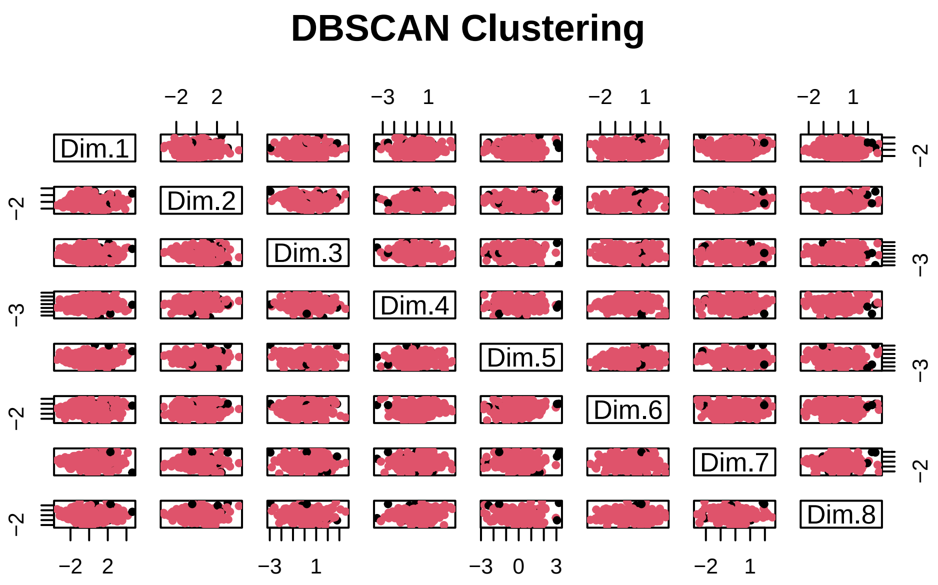
DBSCAN

```
epsilon = 3

dbscan_result <- fpc::dbscan(famd_data, eps = epsilon, MinPts = 4)
# Get cluster labels
cluster_labels <- dbscan_result$cluster

unique_clusters <- unique(cluster_labels)

print(unique_clusters)
```

```
[1] 1 0
```

Remarks: The result above shows that `DBSCAN` failed to identify distinct clusters within the data.

```
plot(famd_data, col = cluster_labels+1, pch = 20, main = "DBSCAN Clustering")
```


DBSCAN Clustering
Dim.1
Dim.2
Dim.3
Dim.4
Dim.5
Dim.6
Dim.7
Dim.8