Exercice 1

1. distX = as.matrix(X) distX = distX ^{2}

2. Mthode 1 XC = scale(X, scale=T) W = XC%*%t(XC)

Mthode 2 QN = diag(nrow(X)) - matrix(1, nrow(X), nrow(X))/nrow(X) W = -1/2*QN%*%distX%*%QN

3. Pour vrifier si elle est dfinie semi-positive, il suffit de vrifier que les valeurs propres soient positives ce qui est le cas car on peut considrer que -1.37*10$^-$17$et - 2.45 * 10^-16 sontdesvaleursnulles.eigen(W)$

4. L = eigen(W)$valuesL = diag(nrow(X)) * L$

V = eigen(W)$vectors$

5. C = V%*%sqrt(L) pas oublier de retirer les NaN

plot(C) idem biplot(princomp(X))

Exercice 2

m = as.vector(mutation) b = cmdscale(mutation, 2, T) c = as.vector(dist(b$points))$plot(b, c) problmemaisonest$p
etc... / sum

on refait de mme avec cmdscale(mutation, 3, T) jusqu' 5

Exercice 3

library(cluster) clusplot

Iris

iris = iris[,1:4] res2 = kmeans(iris, 2) plot(iris, col= res2$cluster)clusplot(iris, res$2cluster)i
res2 = kmeans(iris, 3) ¿ plot(iris, col= res2$cluster$)

2 types diffrents : 143 ou 78.9 pour l'inertie des classes (tot.withinss)

$for(jin2 : 10)for(iin1 : 100)test[j, i] = kmeans(iris, j)$tot.withinss $apply(test, 2, min) Lasolutionquiappar$

Crabs

library(MASS)

Mutations

res = kmeans(mutations2, 2) plot(cmdscale(mutations), col=res$cluster$)

Avec 3 vert au milieu des noirs

4 cluster seulement un point dans le dernier

tableau de contingence pour comparer les partitions table(res$cluster, res$2cluster)