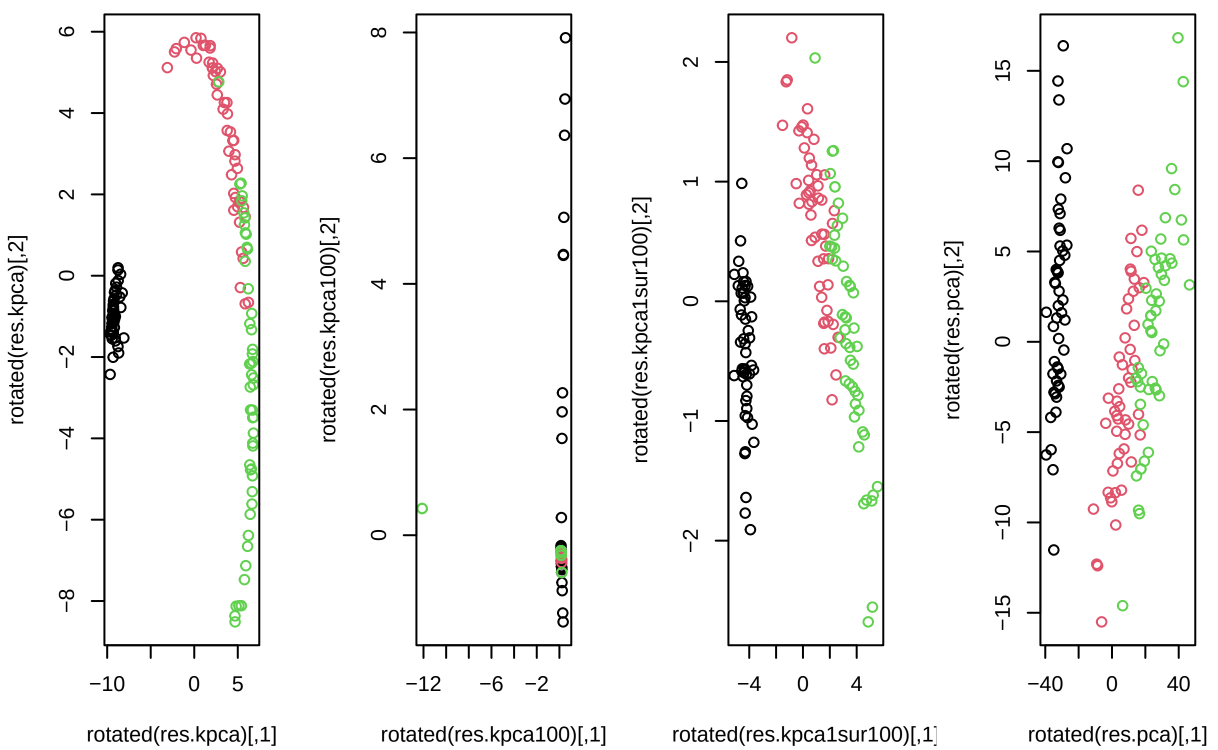
Exercise 1

KPCA with `iris` data set:

`kpca` function kernel:

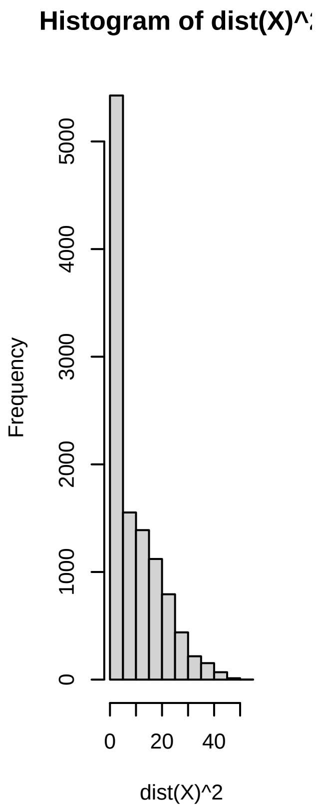
- `vanilladot`: scalar product (Euclidean distance)
- `rbfdot`: $\exp\left(-\frac{1}{2\sigma^2}||x_i - x_j||_2^2\right)$

```
library(kernlab)
X <- iris[, 1:4]
#pairs(X, col=iris$Species)

X <- as.matrix(X)
kpca(X) -> res.kpca
#str(res.kpca)
kpca(X, kpar=list(sigma=100)) -> res.kpca100
kpca(X, kpar=list(sigma=1/100)) -> res.kpca1sur100
kpca(X, kernel="vanilladot", kpar=list()) -> res.pca

par(mfrow=c(1,4))
plot(rotated(res.kpca), col=iris$Species)
plot(rotated(res.kpca100), col=iris$Species)
plot(rotated(res.kpca1sur100), col=iris$Species)
plot(rotated(res.pca), col=iris$Species)
```



```r
hist(dist(X)^2)
summary(dist(X)^2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.090   5.570   9.146  15.050  50.200
```



Programming KPCA

1. $X \to K = (K_{ij})$
2. Calculate $\tilde{K} = K - \frac{2}{n}\mathbb{1}K + \frac{1}{n^2}\mathbb{1}K\mathbb{1}$
3. Diagonalize $\tilde{K}$
4. Project

$$X = UDVXX^T = UDVV^TD^TU^T = UD^2U^T$$

```r
myKPCA <- function(X, k=2, kernel="Gaussian", beta=1) {
  X <- as.matrix(X)
  if (kernel == "Gaussian") {
    K <- exp(-1/beta*(as.matrix(dist(X))^2))
  }
  else {
    K <- X %*% t(X)
```

```
  }

  # Centering
  n <- nrow(X)
  II <- matrix(1/n,n,n)
  Ktilde <- K - 2*II %*% K + II %*% K %*% II

  # Eigenvalue decomposition
  res <- svd(Ktilde)
  alpha <- res$u
  lambda <- res$d^2

  # Projection
  Y <- K %*% alpha[,1:k]

  return(list(Y=Y, lambda=lambda[1:k]))
}
```

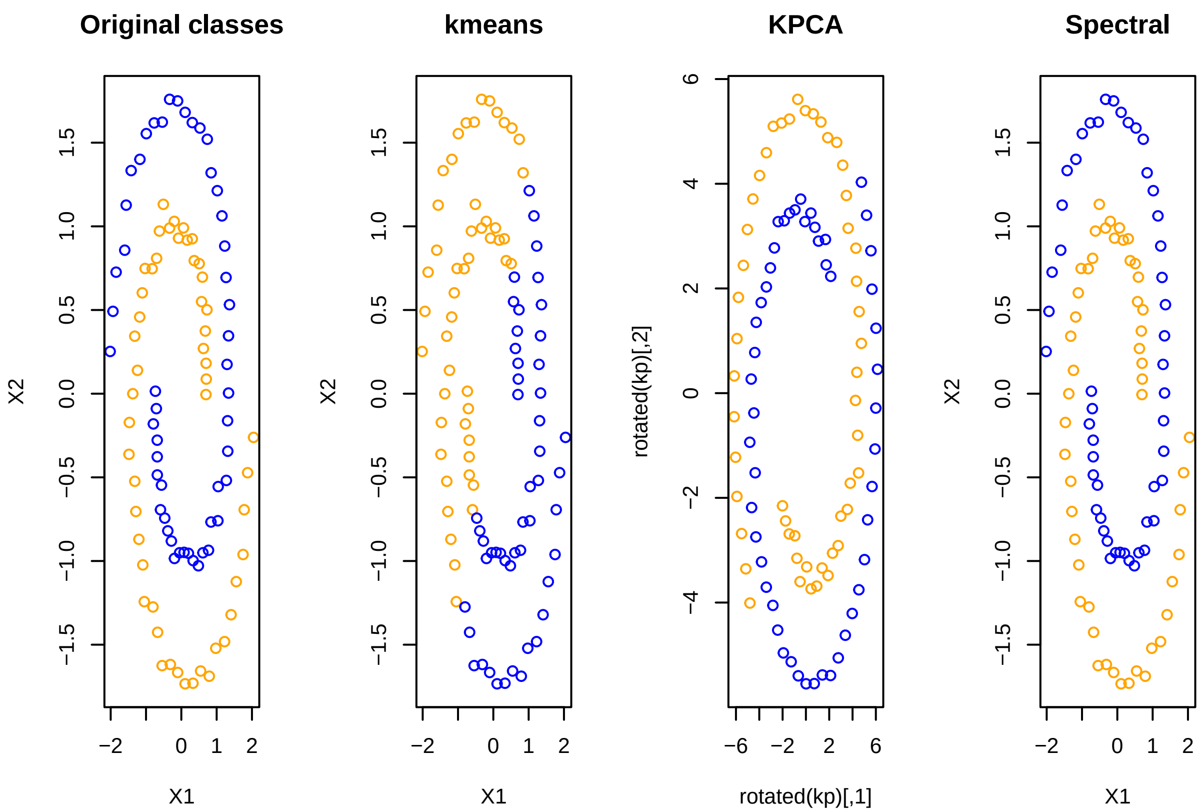
05-12-2022

```
data(spam)
set.seed(1)
train <- sample(1:dim(spam)[1], 400)
kpc <- kpca(~., data=spam[train,-58], kernel="rbfdot", kpar=list(sigma=1/1000), features=2)

#spam %>% PCA(., quali.sup=ncol(spam[train,]))%>% plot(habillage=ncol(spam[train,]), choix="ind")
```

Exercise 2

1.

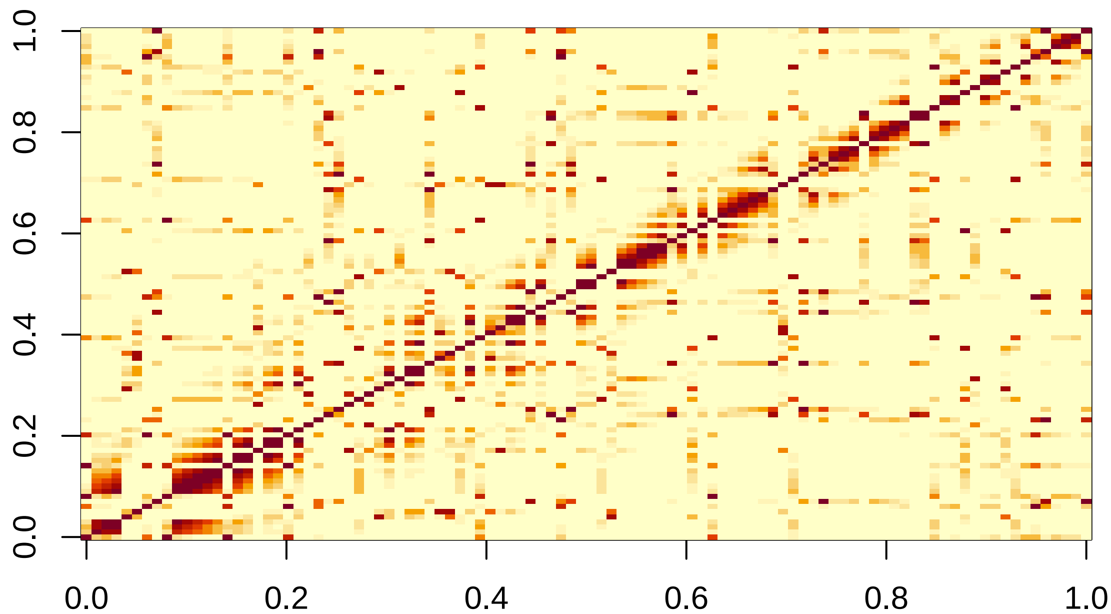
```
library(mlbench)

set.seed(NULL)
obj <- mlbench.spirals(100, 1, 0.025)
my.data <- data.frame(obj$x)
names(my.data) <- c("X1", "X2")
my.data <- scale(my.data, TRUE, TRUE)
par(mfrow=c(1,4))
plot(my.data, col=c("orange", "blue")[obj$classes], main="Original classes")
my.data <- as.matrix(my.data)
plot(my.data, col=c("orange", "blue")[kmeans(my.data, 2)$cluster], main="kmeans")

kpca(my.data, kernel="rbfdot") -> kp
plot(rotated(kp), col=c("orange","blue")[obj$classes], main="KPCA")

specc(my.data, centers=2, kernel="rbfdot") -> sp
plot(my.data, col=c("orange","blue")[sp], main="Spectral")
```



2.

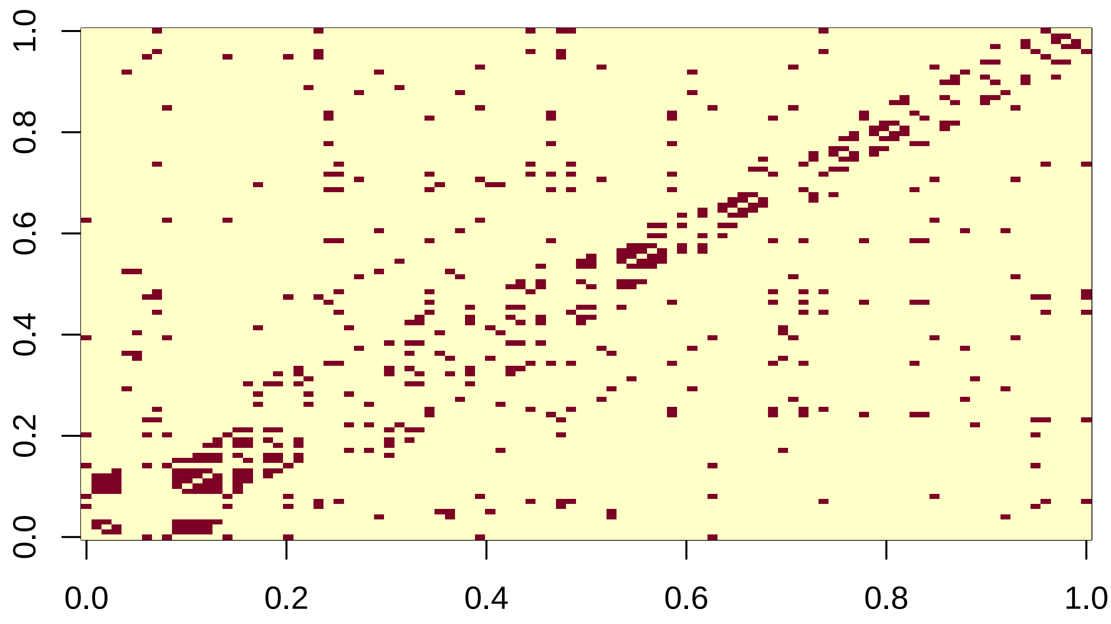
$$K_{ij} = \exp\left(-\frac{1}{\sigma^2}||x_i - x_j||^2\right)$$

```
set.seed(111)
obj <- mlbench.spirals(100, 1, 0.025)
my.data <- data.frame(obj$x)

sigma2 = 1/10
K <- exp(-as.matrix(dist(my.data))^2 / sigma2)
image(K)
```

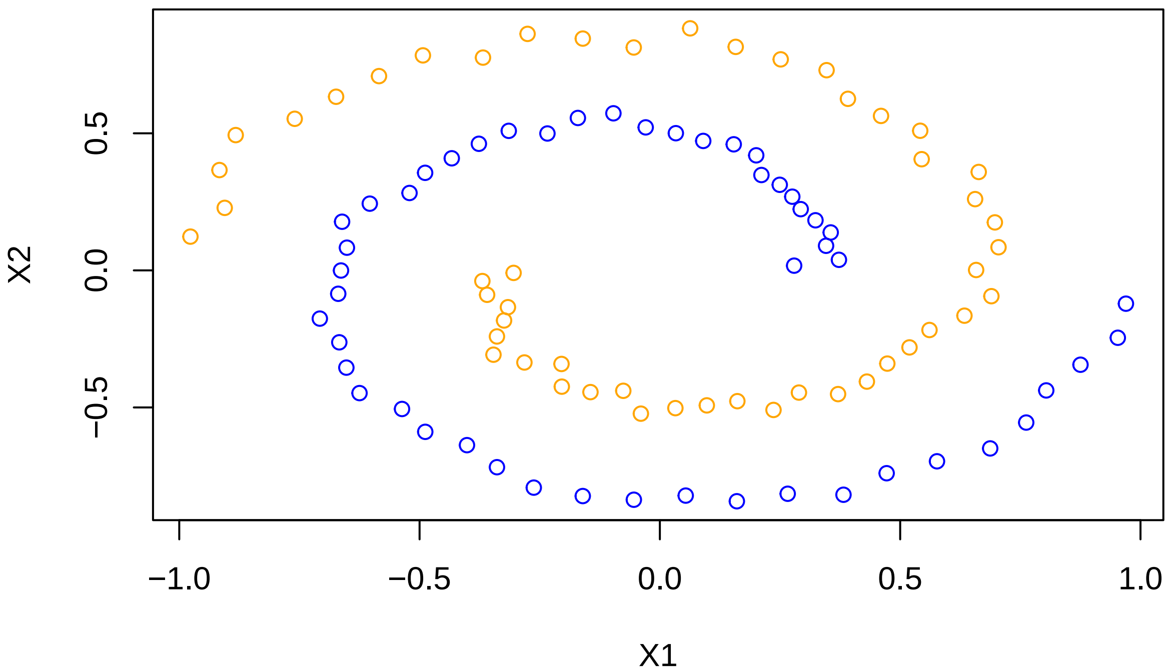


3.

```
A <- K > 0.5
diag(A) <- 0
D <- diag(colSums(A))
L <- D-A
# Li <- diag(rep(1:100)) - solve(D)%*%A
image(A)
```




```
color.kmeans <- kmeans(eigen(L)$vectors[,97:100], 2, nstart=30)$cluster
plot(my.data, col=c("orange","blue")[color.kmeans])
```



Bonus exercise

Show that if λ is eigenvalue of $D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ then $1-\lambda$ is an eigenvalue of $I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$ associated to the same eigenvector.

Solution:

$$L_{(ii)} = I - K$$

Si v est vecteur propre de K

$$Kv = \lambda v - Kv = -\lambda v v(-Kv) = v(-\lambda v)(I-K)v = (1-\lambda)v \Longleftrightarrow$$

$(1-\lambda)$ est vp de $I-K$ associe' 'a v