lab.R

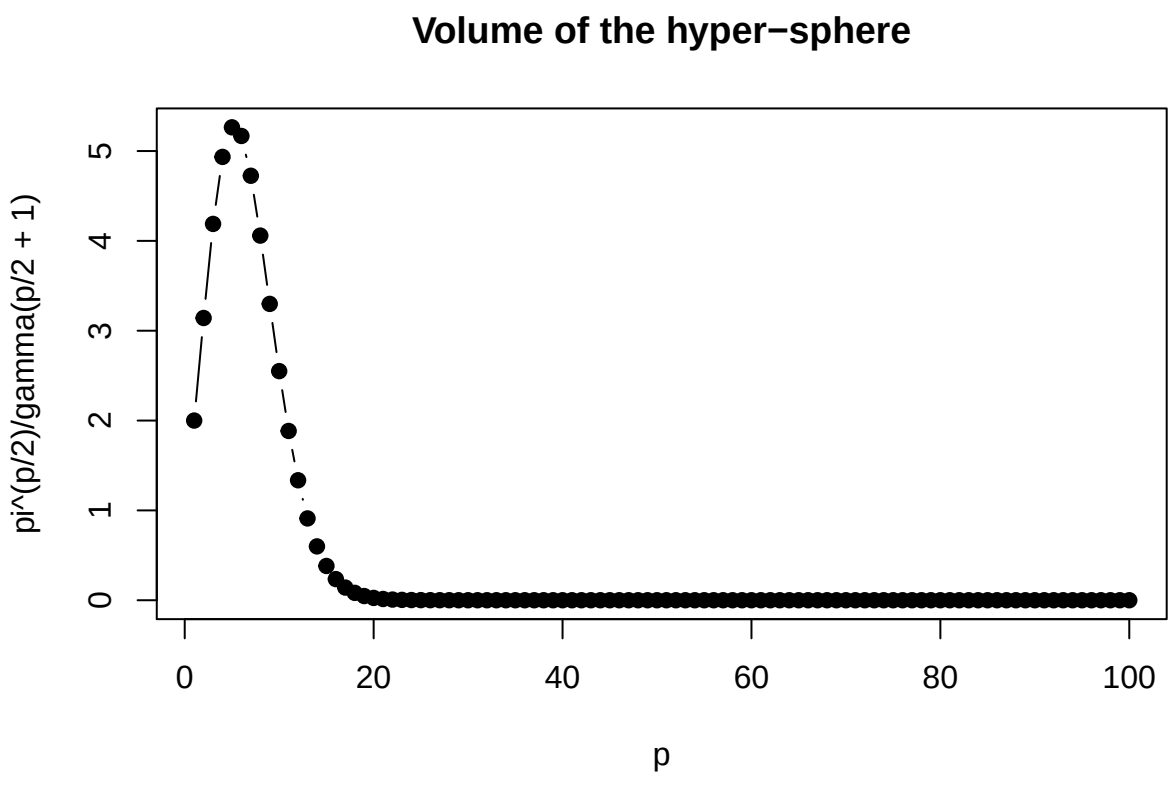
charles

Thu Oct 15 15:06:27 2015

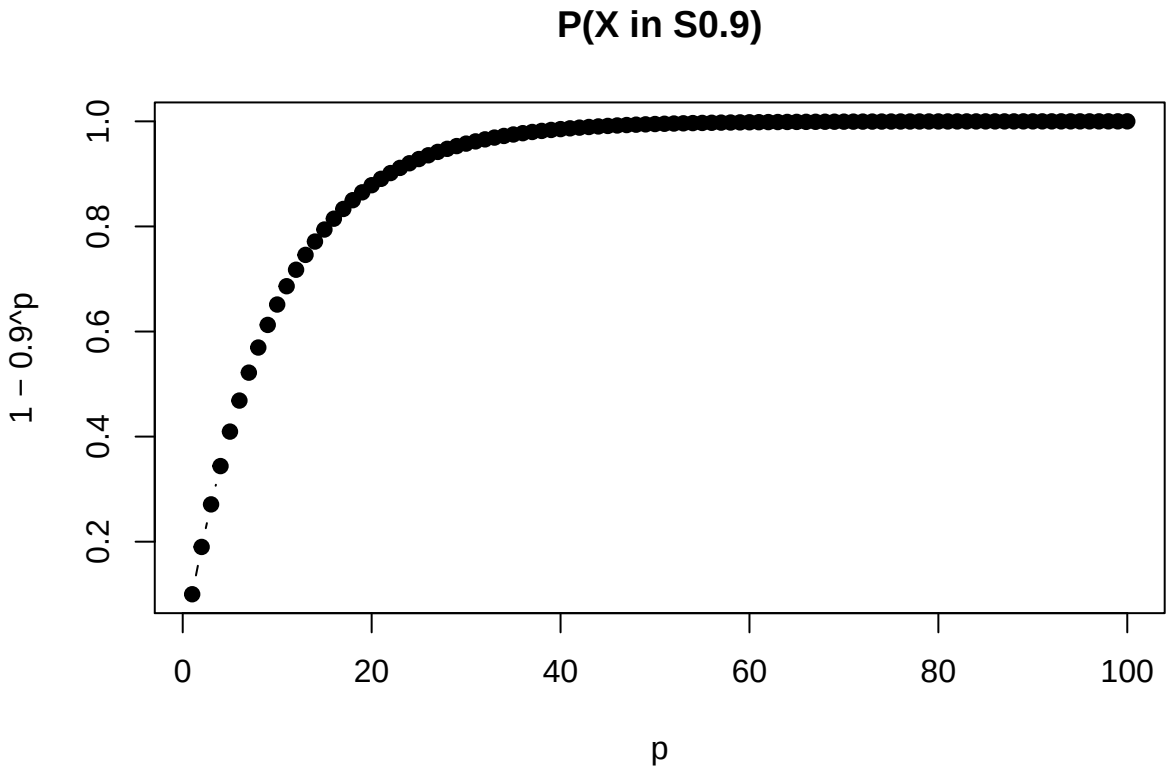
```
myPalette <- c("black","#E41A1C","#377EB8","#4DAF4A","#984EA3","#FF7F00","#FFFF33","#A65628","#F781BF")
palette(myPalette); par(pch=19)
plotIm <- function(x){image(t(matrix(t(x),ncol=16,byrow=TRUE)[16:1,]),col=gray(255:0/255),axes=F); box()

################################################################################
# Loading of data
# load('data/Wine.Rdata')           # n = 150,  p = 13
# load('data/Chironomus.Rdata')     # n = 149, p = 17
# load('data/USPS358.Rdata')        # n = 1756, p = 256
# load('data/Mars.Rdata')           # n = 38400, p = 255
# load('data/Galaxy-small.Rdata')   # n = 10000, p = 1539
# load('data/NIR_data.Rdata')       # n = 202, p = 2800
# also Velib data

################################################################################
# Hyper-sphere and shell
p = 1:100
plot(p,pi^(p/2)/gamma(p/2+1),type='b',main="Volume of the hyper-sphere")
```



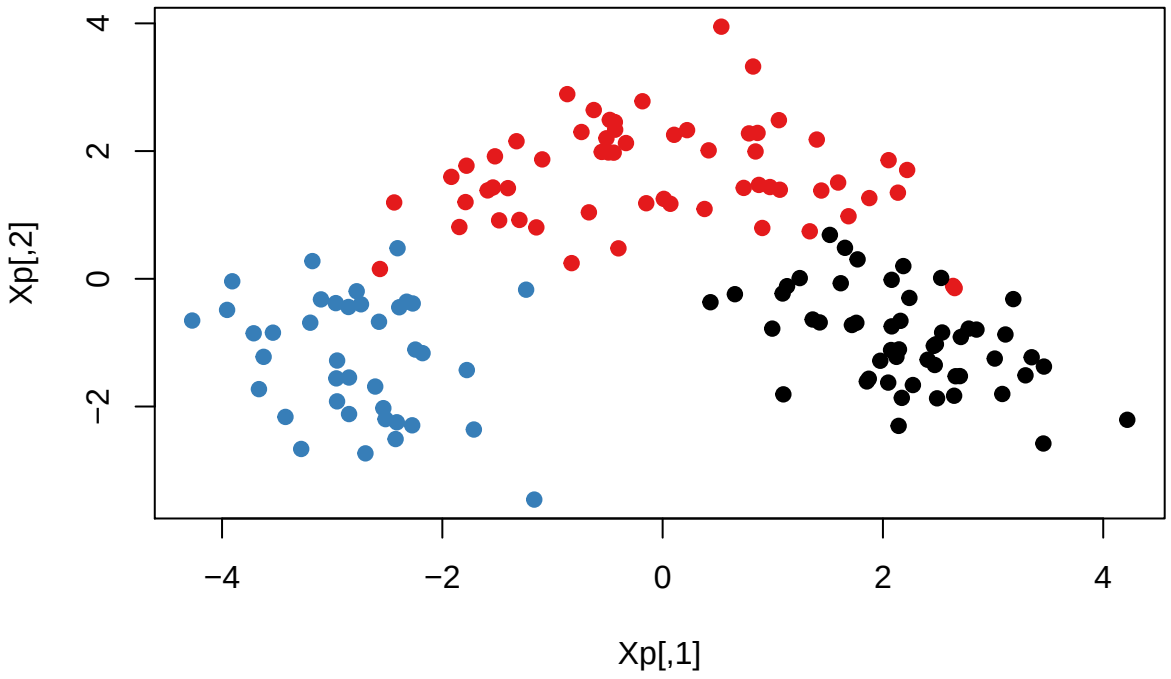

```
plot(p,1-0.9^p,type='b',main='P(X in S0.9)')
```



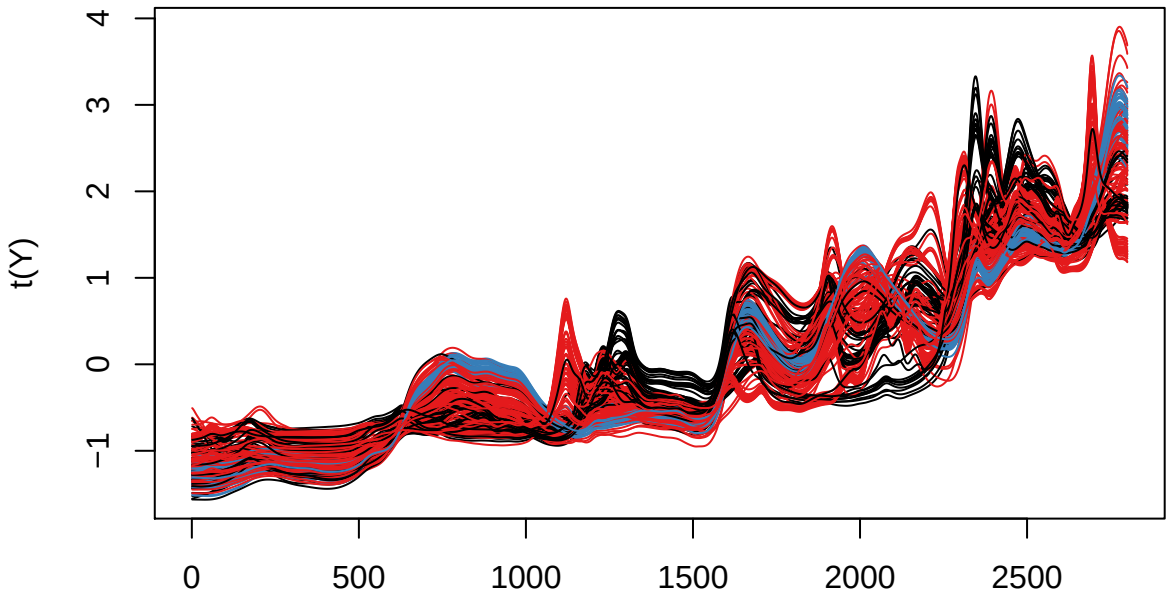

```
################################################################################
# PCA basics
load('data/Wine.Rdata')
pc  = princomp(X)

S = cov(X)
vect = eigen(S)$vect

Xp = predict(pc)[,1:2]
Xp = as.matrix(X) %*% vect[,1:2]
plot(Xp,col=cls)
```

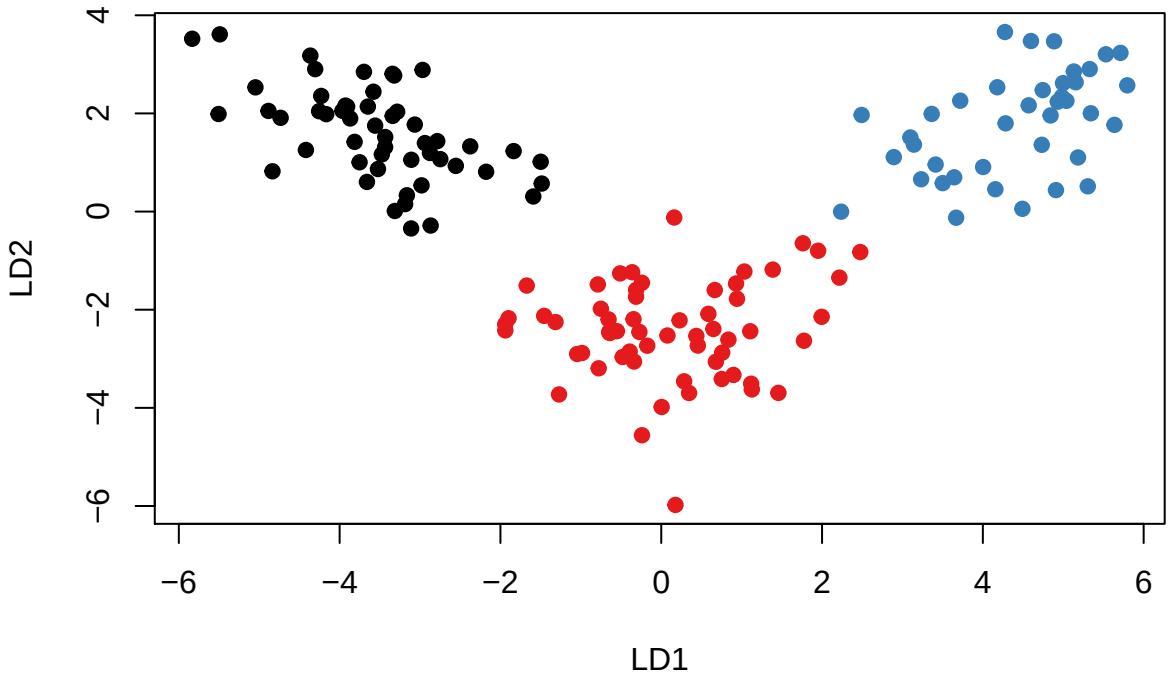



```
################################################################################
# PCA on NIR data
load('data/NIR_data.Rdata')
matplot(t(Y),col=cls,type='l',lty=1)
```




```
#pc = princomp(Y) # does not work since n < p
SS = cov(t(Y)) # PCA through eigen(X %*% t(X))
V = eigen(SS)$vect
U = t(Y) %*% V
Yp = as.matrix(Y) %*% U[,1:10]
################################################################################
# PCA vs FDA on USPS data
```

```
load('data/Wine.Rdata')
library(MASS)
out = lda(X,cls)
Xp = as.matrix(X) %*% out$scaling
plot(Xp,col=cls)
```




```
################################################################################
# GMM with Mclust on Wine and Chironomus
# Try also on NIR data or USPS
library(mclust)
```

```
## Package 'mclust' version 5.0.2
## Type 'citation("mclust")' for citing this R package in publications.
```

```
load('data/Wine.Rdata')
out = Mclust(X,3,modelNames = 'VVV')
adjustedRandIndex(out$class,cls)
```

```
## [1] 0.7348429
```

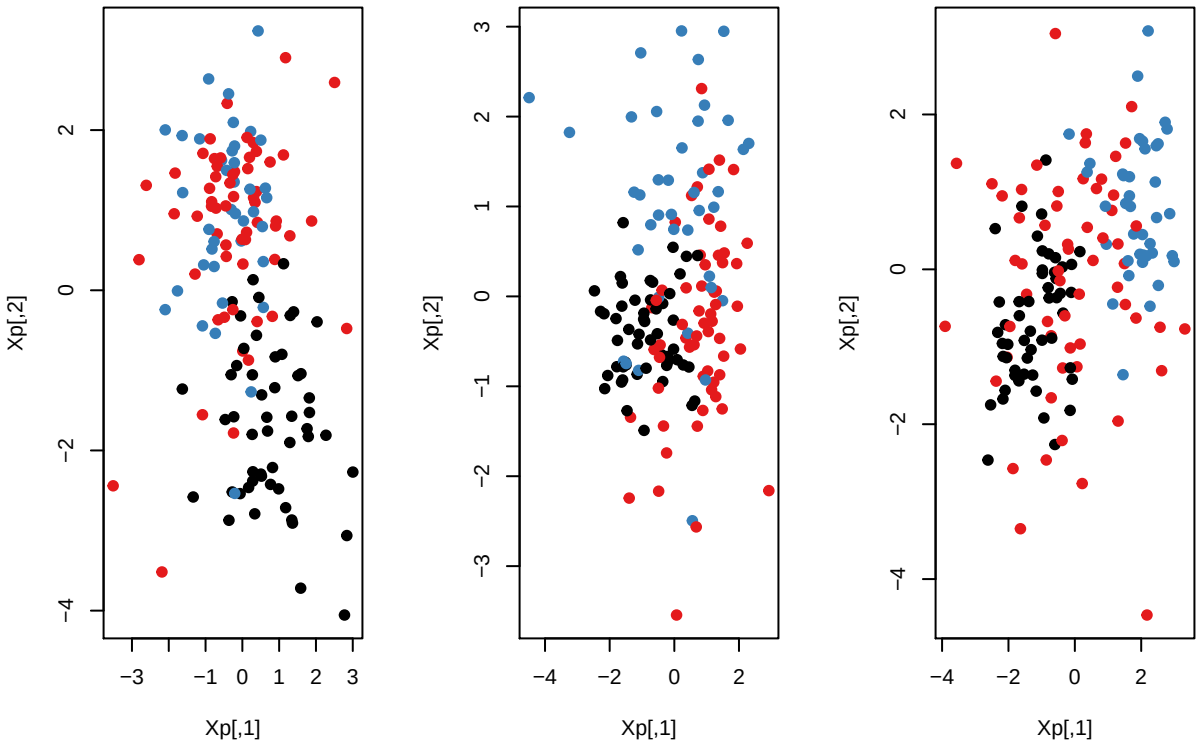
```
################################################################################
# GMM with Mclust on principal components (HOMEWORK)



################################################################################
# HDDC (look at parameters and choice of the dimensionality)
#install.packages('HDclassif')
library(HDclassif)
load('data/Wine.Rdata')
?hddc
out = hddc(X,3)
```

```
##    Model        K       BIC
##   AKJBKQKDK   3   -4918.12
```

```
Q = out$Q
par(mfrow=c(1,3))
for (k in 1:3){
  Qk = Q[[k]]
  Xp = as.matrix(X) %*% Qk[,1:2]
  plot(Xp,col=cls)
}
```




```
library(mclust)
adjustedRandIndex(out$class,cls)
```

```
## [1] 0.935169
```

```
out = hddc(X,3,model='all')
```

```
##    Model        K       BIC
##   AKJBKQKDK   3   -4918.12
##   AKBKQKDK    3   -4978.006
##   ABKQKDK     3   -5004.211
##   AKJBQKDK    3   -4929.188
##   AKBQKDK     3   -4981.316
##   ABQKDK      3   -4838.64
##   AKJBKQKD    3   -4768.343
##   AKBKQKD     3   -4757.561
##   ABKQKD      3   -4753.388
```

```
##   AKJBQKD     3   -4788.418
##   AKBQKD      3   -4777.571
##   ABQKD       3   -4773.411
##   AJBQD       3   -4783.755
##   ABQD        3   -4779.565
##
## SELECTED: model ABKQKD with 3 clusters, BIC=-4753.388.
```

```
adjustedRandIndex(out$class,cls)
```

```
## [1] 0.879818
```

```
out = hddc(X,K=2:7)
```

```
##    Model        K       BIC
##   AKJBKQKDK   2   -4873.308
##   AKJBKQKDK   3   -4918.12
##   AKJBKQKDK   4   -4951.246
##   AKJBKQKDK   5   -5240.134
##   AKJBKQKDK   6   -5204.105
##   AKJBKQKDK   7   -5320.495
##
## SELECTED: model AKJBKQKDK with 2 clusters, BIC=-4873.308.
```

```
adjustedRandIndex(out$class,cls)
```

```
## [1] 0.3601701
```

```
################################################################################
# FisherEM (look at parameters and visualization)
#install.packages('FisherEM')
library(FisherEM)
```

```
## Loading required package: elasticnet
## Loading required package: lars
## Loaded lars 1.2
```

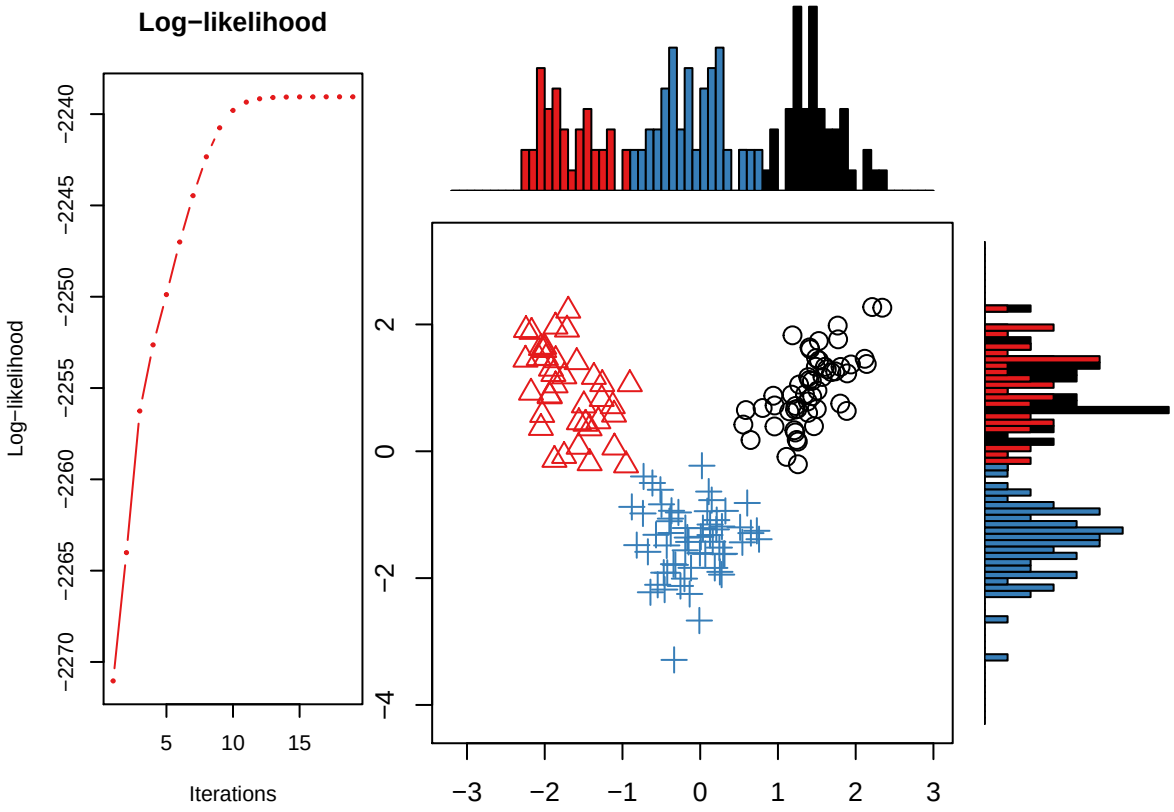
```
?fem
out = fem(X,3)
adjustedRandIndex(out$cls,cls)
```

```
## [1] 0.879615
```

```
out = fem(X,3,model='all')
adjustedRandIndex(out$cls,cls)
```

```
## [1] 0.9616115
```

```
plot(out)
```




```
################################################################################
# SparseFEM
out = sfem(X,3,l1 = 0.4,model='all')
adjustedRandIndex(out$cls,cls)
```

```
## [1] 0.9616115
```

```
out$U
```

```
##                U1         U2
## V2  -7.622002e-01  0.0000000
## V3   0.000000e+00  0.0000000
## V4   0.000000e+00  0.0000000
## V5   0.000000e+00  0.0000000
## V6   0.000000e+00  0.0000000
## V7   0.000000e+00  0.0000000
## V8  -1.110223e-16 -0.9852941
## V9   0.000000e+00  0.0000000
## V10  0.000000e+00  0.0000000
## V11 -6.434946e-01  0.0000000
## V12  0.000000e+00  0.0000000
## V13  0.000000e+00 -0.1708669
## V14 -7.046677e-02  0.0000000
```

```
plot(out)
```