protein_analisys.R

ra016120

Fri Feb 19 17:09:49 2016

```
## Carregando os pacotes necessários
library(readr)

## Lendo o conjunto de dados
protein <- read.table('protein.txt', header = T, sep = '\t')

## Sumário básico
summary(protein)
```

```
##            Country      RedMeat         WhiteMeat           Eggs
##  Albania       : 1   Min.   : 4.400   Min.   : 1.400   Min.   :0.500
##  Austria       : 1   1st Qu.: 7.800   1st Qu.: 4.900   1st Qu.:2.700
##  Belgium       : 1   Median : 9.500   Median : 7.800   Median :2.900
##  Bulgaria      : 1   Mean   : 9.828   Mean   : 7.896   Mean   :2.936
##  Czechoslovakia: 1   3rd Qu.:10.600   3rd Qu.:10.800   3rd Qu.:3.700
##  Denmark       : 1   Max.   :18.000   Max.   :14.000   Max.   :4.700
##  (Other)       :19
##       Milk            Fish           Cereals          Starch
##  Min.   : 4.90   Min.   : 0.200   Min.   :18.60   Min.   :0.600
##  1st Qu.:11.10   1st Qu.: 2.100   1st Qu.:24.30   1st Qu.:3.100
##  Median :17.60   Median : 3.400   Median :28.00   Median :4.700
##  Mean   :17.11   Mean   : 4.284   Mean   :32.25   Mean   :4.276
##  3rd Qu.:23.30   3rd Qu.: 5.800   3rd Qu.:40.10   3rd Qu.:5.700
##  Max.   :33.70   Max.   :14.200   Max.   :56.70   Max.   :6.500
##
##       Nuts           Fr.Veg
##  Min.   :0.700   Min.   :1.400
##  1st Qu.:1.500   1st Qu.:2.900
##  Median :2.400   Median :3.800
##  Mean   :3.072   Mean   :4.136
##  3rd Qu.:4.700   3rd Qu.:4.900
##  Max.   :7.800   Max.   :7.900
##
```

```
## Mudando a escala
pmatrix <- scale(protein[,-1])
pcenter <- attr(pmatrix, "scaled:center")
pscale <- attr(pmatrix, "scaled:scale")

## Criando o cluster
d <- dist(pmatrix, method="euclidean")
pfit <- hclust(d, method="ward")
```

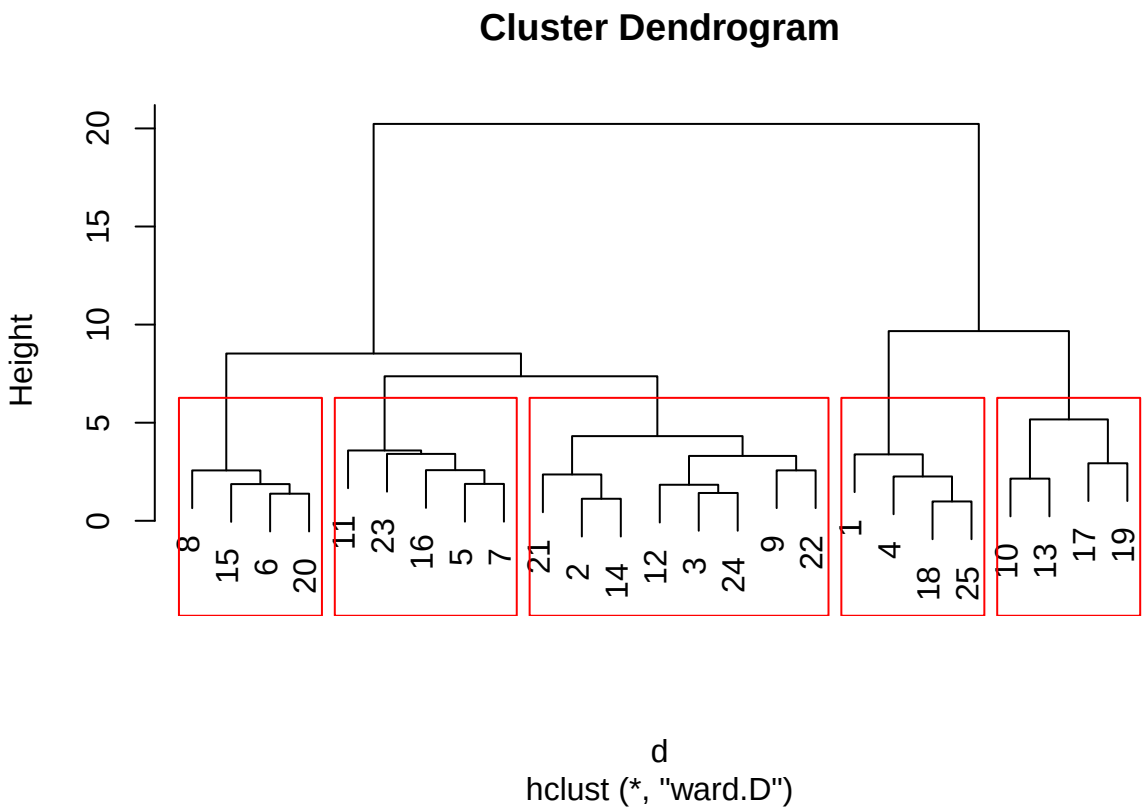
```
## The "ward" method has been renamed to "ward.D"; note new "ward.D2"
```

```r
plot(pfit)

## Desenhando a separação dos clusters
rect.hclust(pfit, k=5)

## Obtendo os grupos para cada cluster
groups <- cutree(pfit, k=5)

## Função para imprimir os cluster
print_clusters <- function(labels, k) {
  for(i in 1:k) {
    print(paste("cluster", i))
    print(protein[labels==i,c("Country","RedMeat","Fish","Fr.Veg")])
  }
}

## Imprimindo os cluster
print_clusters(groups, 5)
```

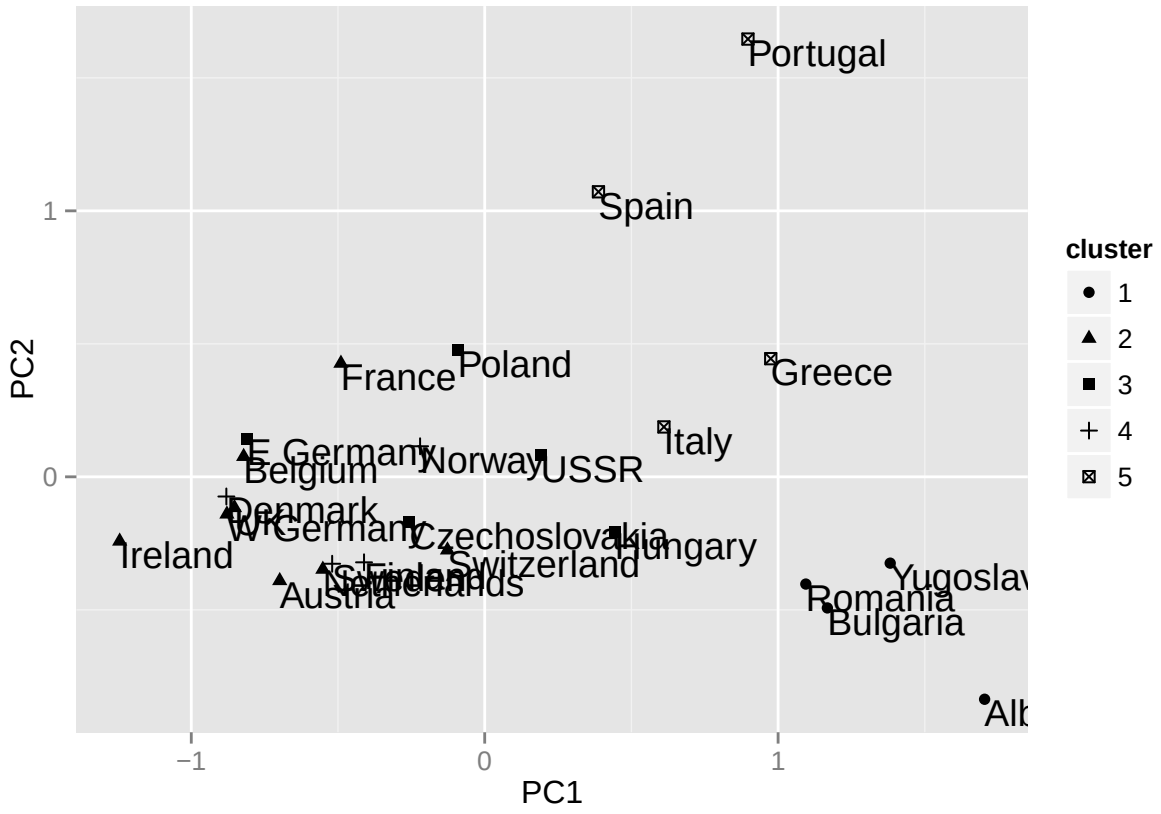
```
## [1] "cluster 1"
##       Country RedMeat Fish Fr.Veg
## 1     Albania    10.1  0.2    1.7
## 4    Bulgaria     7.8  1.2    4.2
## 18    Romania     6.2  1.0    2.8
## 25 Yugoslavia     4.4  0.6    3.2
## [1] "cluster 2"
##        Country RedMeat Fish Fr.Veg
## 2      Austria     8.9  2.1    4.3
## 3      Belgium    13.5  4.5    4.0
## 9       France    18.0  5.7    6.5
## 12     Ireland    13.9  2.2    2.9
## 14 Netherlands     9.5  2.5    3.7
## 21 Switzerland    13.1  2.3    4.9
## 22          UK    17.4  4.3    3.3
## 24   W Germany    11.4  3.4    3.8
## [1] "cluster 3"
##           Country RedMeat Fish Fr.Veg
## 5  Czechoslovakia     9.7  2.0    4.0
## 7       E Germany     8.4  5.4    3.6
## 11        Hungary     5.3  0.3    4.2
## 16         Poland     6.9  3.0    6.6
## 23           USSR     9.3  3.0    2.9
## [1] "cluster 4"
##    Country RedMeat Fish Fr.Veg
## 6  Denmark    10.6  9.9    2.4
## 8  Finland     9.5  5.8    1.4
## 15  Norway     9.4  9.7    2.7
## 20  Sweden     9.9  7.5    2.0
## [1] "cluster 5"
##     Country RedMeat Fish Fr.Veg
## 10   Greece    10.2  5.9    6.5
## 13    Italy     9.0  3.4    6.7
## 17 Portugal     6.2 14.2    7.9
```

```
## 19    Spain     7.1  7.0    7.2
```

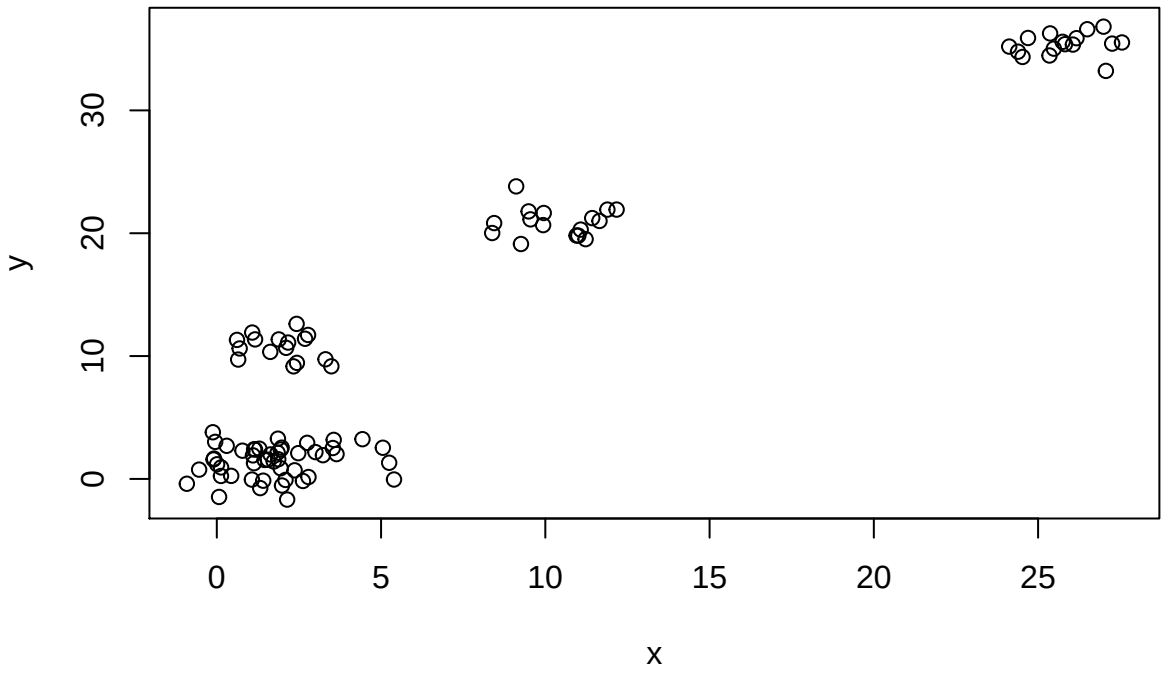
```
## Visualizando os clusters
library(ggplot2)
```



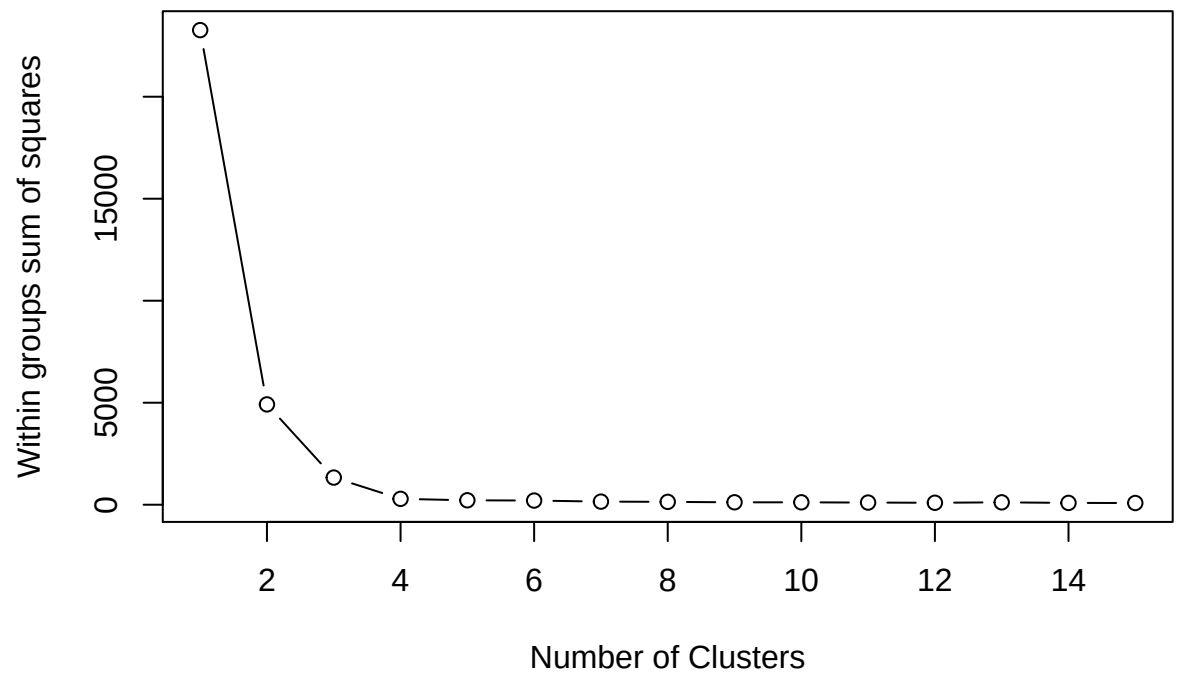

```
princ <- prcomp(pmatrix)
nComp <- 2
project <- predict(princ, newdata = pmatrix)[,1:nComp]

## Criando uma tabela com os dados transformados
project.plus <- cbind(as.data.frame(project),
                      cluster=as.factor(groups),
                      country=protein$Country)

## Fazendo o gráfico
library(ggplot2)
ggplot(project.plus, aes(x=PC1, y=PC2)) +
  geom_point(aes(shape=cluster)) +
  geom_text(aes(label=country), hjust=0, vjust=1)
```




```
## Fazendo em 3D
## Fazendo o plot em 3D
## Projetando em 3 dimensões
project <- predict(princ, newdata = pmatrix)[,1:3]
library(rgl)
plot3d(project, col=project.plus$cluster, pch = 19)

## Hack para fazer em 3D algo similar ao que foi feito
plot3d(project, col=project.plus$cluster, pch = 19)
text3d(project,texts=project.plus$country)

###################### K-MEANS ######################################
## Gerando um conjunto de dados fictício
n = 100
g = 6
set.seed(g)
d <- data.frame(x = unlist(lapply(1:g, function(i) rnorm(n/g, runif(1)*i^2))),
                y = unlist(lapply(1:g, function(i) rnorm(n/g, runif(1)*i^2))))
plot(d)
```




```
## Método do cotovelo
mydata <- d
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(mydata,
                                     centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```



```r
## Aplicando o k-means e o método do cotovelo para os dados
## da proteína
mydata <- pmatrix
wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(mydata,
                                     centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares")
```