```
#
#
# CÓDIGO
#
# Este código explora o data frame USArrests (datasets) empregando diferentes técnicas
# de clusterização, métodoso de avaliação dos grupos, visualização dos dados etc.
#
# O código está dividido nas seguintes seções:
#
# 1. PREPARE CONHEÇA SEU DADO
# 2. BUSCANDO O MELHOR K NÚMERO DE GRUPOS
# 2.1. Within and Betweenss: Elbow
# 2.2. Group Sizes
# 2.3. Silhouette Method
# 3. ANALISE DOS GRUPOS (DOS RESULTADOS)
# 4. PAM K MEDÓIDES
# 5. CLUSTER HIERARQUICO HCLUST
# 6. OUTROS MÉTODOS E DISCUSSÃO
#
# A cada seção do código, explicadas neste texto de apoio, execute e explore o código,
# e responda as questões correspondente.
#



# ------------------------------------------------------------------------------
# IMPORTANTE:
# Muitos processos aqui executados empregam técnicas de resampling.
# No caso de estar obtendo resultados diferentes das alternativas do
# questionário, limpe todas as variáveis (o workspace do RStudio)
# e re-execute o trecho em questão. Não esqueça também de manter o seed.
# ------------------------------------------------------------------------------
```

```
#
# Um ou mais pacotes podem ser necessários. Instale a medida que forem
# necessários
# ------------------------------------------------------------------------------

# install.packages("datasets")
# install.packages("cluster")
# install.packages("tidyverse")
# install.packages("corrplot")
# install.packages("gridExtra")
# install.packages("GGally")
 install.packages("factoextra")
# if(!require(devtools)) install.packages("devtools")
# devtools::install_github("kassambara/factoextra")

library(datasets) # USArrests
library(cluster)
library(tidyverse)
library(corrplot)
library(gridExtra)
library(GGally)
library(factoextra)

#
# 1. PREPARE CONHEÇA SEU DADO                    Q1
# ------------------------------------------------------------------------------

#
# Know your data before start
```

```
#
help(USArrests)
head(USArrests)
summary(USArrests)

#
# Missing Data: Dados faltantes devem ter tratamento antes da clusterização
#
# Há dados faltantes aqui? R - Não
#
any(is.na(USArrests))

#
# Numeric Data: Como a Clusterização emprega funções de similaridade/distância somente
# dados numéricos são aceitos
#
# Há dados NãO numéricos aqui? R - Não
#
any(as.logical(sapply(USArrests, is.numeric )-1))

#
# Scale data: diferentemente dos processos de classificação (aprendizado supervisionado)
# é importante aplicar-se algum tipo normalização os dados dos dados
#
# A função scale( ) normaliza todos os valores com centro=0 (media) e variancia=1
#
mydata = data.frame(scale(USArrests))
summary(mydata)
sapply(mydata,median) # cuidado com o nr de zeros!
sapply(mydata,var)
```

```
#
# 2. BUSCANDO O MELHOR K NÚMERO DE GRUPOS
# ------------------------------------------------------------------------------


#
# 2.1. Within and Betweenss: Elbow              Q2
# ------------------------------------------------------------------------------

set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')

wss = 0
bss = 0
for (i in 1:10) wss[i] = sum(kmeans(mydata,i, nstart=25)$withinss)
for (i in 1:10) bss[i] = sum(kmeans(mydata,i, nstart=25)$betweenss)

par(mfrow=c(1,2))
plot(1:10, wss, type="b", xlab="Number of Clusters", ylab="Within groups sum of squares")
# Voce só poderá definir o valor abaixo depois de inspecionar no gráfico acima
# o ponto de cotovelo
abline(v=4,col="red")

plot(1:10, bss, type="b", xlab="Number of Clusters", ylab="Betweenss groups sum of squares")
# Voce só poderá definir o valor abaixo depois de inspecionar no gráfico acima
# o ponto de cotovelo
abline(v=4,col="red")

sum(wss)
sum(bss)

#
```

```
# consulte o help(kmeans) e empregue a função para responder a Q2
 fit = kmeans(mydata, 4, nstart=25)
 fit$withinss
#


#
# 2.2. Group Sizes                          Q3
# ------------------------------------------------------------------------------

set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')

par(mfrow=c(2,2))
for (i in 3:6){
  fit <- kmeans(mydata,i, nstart=25)
  main_ = paste("Size groups for k=", i)
  barplot(fit$size, main = main_ )
}

#
# consulte o help(kmeans) e empregue a função para responder a Q3
 fit = kmeans(mydata, 4, nstart=25)
 fit
#


#
# 2.3. Silhouette Method                    Q4
# ------------------------------------------------------------------------------

set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')
```

```
ss_m = c(0)
for (i in 3:10){
  fit <- kmeans(mydata,  i, nstart=25)
  ss <- silhouette(fit$cluster, dist(mydata))
  ss_m[i] <-  mean(ss[,3])
}

par(mfrow=c(1,1))
plot(ss_m,
    type = "b", pch = 19, frame = FALSE,
    xlab = "Number of clusters K",
    ylab = "Average Silhouettes",
    xlim=c(3,10))
# Voce só poderá definir o valor abaixo depois de inspecionar no gráfico acima
# o ponto de cotovelo
abline(v=4,col="red")

set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')

par(mfrow=c(2,2))
fit <- kmeans(mydata,  3, nstart=25)
ss <- silhouette(fit$cluster, dist(mydata))
plot(ss)

fit <- kmeans(mydata,  4, nstart=25)
ss <- silhouette(fit$cluster, dist(mydata))
plot(ss)

fit <- kmeans(mydata,  5, nstart=25)
```

```
ss <- silhouette(fit$cluster, dist(mydata))
plot(ss)

fit <- kmeans(mydata,  6, nstart=25)
ss <- silhouette(fit$cluster, dist(mydata))
plot(ss)

# se necessário consulte o help(silhouette) e empregue a função para responder a Q3



#
# 3. ANALISE DOS GRUPOS (DOS RESULTADOS)          Q5
# ------------------------------------------------------------------------------

set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')

fit <- kmeans(mydata,4, nstart=25)

par(mfrow=c(1, 1))
ss <- silhouette(fit$cluster, dist(mydata))
plot(ss)

#
# a. Visualizando o Cluster para os Dois Componentes Principais
# (você vai conhecer mais sobre componente principais na próxima trilha)
#

par(mfrow=c(1, 1))
clusplot(mydata, fit$cluster, color=TRUE, shade=TRUE, labels=2, lines=0)
```

```
#
# você optar pela função mais simples "plotcluster"
# no caso de problema com os pacotes de visualização
#
# library(fpc)
# plotcluster(mydata, fit$cluster)

#
# b. Visualizando o Cluster para Duas variáveis de sua escolha
#

p1 = ggplot(mydata, aes(UrbanPop, Murder, color = as.factor(fit$cluster))) + geom_point()
p2 = ggplot(mydata, aes(UrbanPop, Assault, color = as.factor(fit$cluster))) + geom_point()
p3 = ggplot(mydata, aes(UrbanPop, Rape, color = as.factor(fit$cluster))) + geom_point()

grid.arrange(p1, p2, p3)

USArrests[USArrests$Murder == 0.8,]

USArrests['Florida',]

min(USArrests$Murder)

#
# c. Visualizando os valores médios de cada Cluster
#

mydata$predict = fit$cluster

par(mfrow=c(2, 2))
for (i in 1:4){
```

```
  main_ = paste("Group =", i)
  barplot(sapply(mydata[mydata$predict==i,-5],mean),main=main_)
}

#
# d. Visualizando o Cluster para a diferentes pares de variáveis e suas distribuições
#

ggpairs(cbind(mydata, Cluster=as.factor(fit$cluster)),
        columns=1:4, aes(colour=Cluster, alpha=0.5),
        lower=list(continuous="points"),
        upper=list(continuous="blank"),
        axisLabels="none", switch="both")

#
# 4. PAM K MEDÓIDES                    Q6
# -----------------------------------------------------------------------------

set.seed(1987) # não altere para que seu resultados correspondam ao questionário
RNGversion('3.5.0')

fit = pam(mydata,4)

# Compare PAM e Kmeans
table(fit$cluster, mydata$predict)

ss <- silhouette(fit$cluster, dist(mydata))

par(mfrow=c(1, 1))
plot(ss)
```

```
#
# 5. CLUSTER HIERARQUICO HCLUST          Q7 Q8 Q9
# ----------------------------------------------------------------------------

set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')

d <- dist(mydata, method = "euclidean") # distance matrix

#    ATENÇÃO: altere o código para escolha de um método e execute um método por
#    vez escolhendo o melhor particionamento
#

# fit <- hclust(d, method = "complete")
# fit <- hclust(d, method = "single")
 fit <- hclust(d, method = "average")

#
#    A seguir repetem-se os mesmos procedimentos empregados para o K-médias:
#    análise do Silhouette, Visualização dos resultados e um comparativo com o PAM.

par(mfrow=c(1, 1))
plot(fit) # display dendogram

# cut tree into 4 clusters
groups <- cutree(fit, k=4)

# draw dendogram with red borders around the 4 clusters
rect.hclust(fit, k=4, border="red")

#
```

```
#  Silhouette
#
ss = silhouette(groups, dist(mydata))
plot(ss)


#
# Compare aqui agrupamentos (PAM, hclust) e (Kmeans, hclust) a
# exemplo código acima em que comparamos (PAM, Kmeans)
#
# responda então a questão 9
#
set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')



#
# see results like kmeans
#
par(mfrow=c(1, 1))
clusplot(mydata, groups, color=TRUE, shade=TRUE,
       labels=2, lines=0)

# Centroid Plot against 1st 2 discriminant functions
# library(fpc)
# plotcluster(mydata, groups)

ggpairs(cbind(mydata, Cluster=as.factor(groups)),
      columns=1:4, aes(colour=Cluster, alpha=0.5),
      lower=list(continuous="points"),
      upper=list(continuous="blank"),
      axisLabels="none", switch="both")
```

```
#
# 6. OUTROS MÉTODOS E DISCUSSÃO          Q10
#
# Acesse help(fviz_nbclust) ou pesquise na internet sobre esse pacote
# e execute o código abaixo para os métodos kmeans e hclust
#


set.seed(1987) # não altere para que seu resultados correspondam ao
questionárioset.seed(1987) # não altere para que seu resultados correspondam ao
question?rio
RNGversion('3.5.0')


fviz_nbclust(mydata, kmeans, method = "silhouette")


# Veja no help(fviz_nbclust) o que a função abaixo faz e como alterar o
# parametro de kmeans para o cluster hierarquico (não é hclust! ;-) )


set.seed(1987) # não altere para que seu resultados correspondam ao question?rio
RNGversion('3.5.0')


fviz_nbclust(mydata, FUNcluster = kmeans, method = c("silhouette", "wss",
                                    "gap_stat"), diss = NULL, k.max = 20, nboot = 100,
        verbose = interactive(), barfill = "steelblue", barcolor = "steelblue",
        linecolor = "steelblue", print.summary = TRUE)
```