# R script: “trajetórias de reguladores”

#Defining the working directory

setwd("C:/Users/r1372845/Documents/Trajetórias de reguladores")

seqreg <- read.csv(file="Base reguladores novos codigos sem mandato exercicio reduz2vs1.csv", header=TRUE, sep=";", dec=",", fill=TRUE, na.strings="")

frequencia <- tidyr::gather(seqreg , "Tempo" , "Estado" , 2:11)

nomes\_ord <- names(sort(table(frequencia$Estado),decreasing = TRUE))

nomes\_ord

library(TraMineR)

seqstatl(seqreg, 2:11)

#definindo os estados e labels”

seqreg.labels <- c(

"servidor: político" ,

"servidor" ,

"privado: político" ,

"privado" ,

"não aplicável"

)

seqreg.scode <- c(

"2Spub" ,

"2Npub" ,

"2Spriv" ,

"2Npriv" ,

"n\_aplicavel"

)

seqreg.seq <- seqdef(seqreg, 2:11,

                     states = seqreg.scode,

                     labels = seqreg.labels , alphabet = seqreg.scode)

seqtab(seqreg.seq)

#as sequências mais comuns

x11()

par(mfrow=c(1,2))

seqfplot(seqreg.seq, with.legend = T,

         main = "As 10 sequências mais comuns")

#a distribuição dos estados, ano a ano

x11()

par(mfrow=c(1,2))

seqdplot(seqreg.seq, with.legend = T,

         main = "A distribuição dos estados, ano a ano")

#iniciando o pareamento

submat <- seqsubm(seqreg.seq, method = "TRATE")

dist.om1 <- seqdist(seqreg.seq, method = "OM",indel=1,sm=submat)

library(cluster)

#Ward cluster

clusterward1 <- agnes(dist.om1, diss = TRUE, method = "ward")

#plotando o dendograma

plot(clusterward1, ask=TRUE)

#se quisermos fazer os retângulos…

#um exemplo com 4

plot(as.dendrogram(clusterward1))

rect.hclust(clusterward1 , k = 4)

#Decisão do número de clusters

totss <- function(dmatrix) {

  grandmean <- apply(dmatrix , 2 , FUN=mean)

  sum(apply(dmatrix, 1, FUN = function(row) { sqr\_edist(row,grandmean)}))

}

sqr\_edist <- function(x,y) {

  sum((x-y)^2)

}

wss.cluster <- function(clustermat) {

  c0 <- apply(clustermat , 2 , FUN = mean)

  sum(apply(clustermat , 1 , FUN = function(row) {sqr\_edist(row,c0)}))

}

wss.total <- function(dmatrix,labels){

  wsstot <- 0

  k <- length(unique(labels))

  for (i in 1:k)

    wsstot <- wsstot + wss.cluster(subset(dmatrix, labels==i))

  wsstot

}

#  Distance matrix

submat <- TraMineR::seqsubm(seqreg.seq, method = "TRATE")

dist.om1 <- TraMineR::seqdist(seqreg.seq, method = "OM",indel=1,sm=submat)

##### ----- Codigos de dentro da função, ao ar livre ----- #####

kmax = 10

dmatrix = dist.om1

npts <- dim(dmatrix)[1]  # numero de linhas.

totss2 <- totss(dmatrix)

wss <- numeric(kmax)

cirt <- numeric(kmax)

wss[1] <- (npts-1)\*sum(apply(dmatrix, 2 , var))

# Retirei essas duas linhas de dentro do loop

# Para utilizar hclust -----> d <- as.dist(dmatrix)

pfit <- cluster::agnes(dist.om1 , diss = TRUE , method = "ward")

# base mista #fastcluster::hclust(d , method = "ward.D2")

# Calculando soma de quadrados para todos os numeros de clusters.

for (k in 2:kmax){

  labels <- cutree(pfit , k = k)

  wss[k] <- wss.total(dmatrix , labels)

}

bss <- round( totss2 - wss , 8 )

crit.num <- bss/(0:(kmax-1))

crit.denom <- wss/(npts - 1:kmax)

# Criando lista com indices ch e wss

lista <- list(crit = crit.num/crit.denom ,

              wss  = wss,

              totss = totss)

## --- Calculo do silhouette --- ##

cl  <- list()

sil <- list()

clw <- cutree(pfit , k = 1)

cl[[1]] <- cluster::silhouette( clw , dist.om1 )

sil[[1]] <- cl[[1]]

for (i in 2:kmax){

  clw <- cutree(pfit , k = i)

  cl[[i]]  <- cluster::silhouette( clw , dist.om1 )

  sil[[i]] <- summary(cl[[i]])[[1]][[4]]

}

##### ----- Criacao do data frame com as medidas ----- #####

critframe <- data.frame( k = 1:10 ,

                         ch  = scale(lista$crit),

                         wss = scale(lista$wss),

                         sil = scale(unlist(sil)))

##### ----- Ajustando data.frame para o plot ----- #####

critframe <- reshape2::melt(critframe , id.vars = c("k"),

                            variable.name = "Measure",

                            value.name = "score")

# Plotando o grafico

library(ggplot2)

ggplot(critframe , aes(x = k , y= score , color = Measure)) +

  geom\_point(aes(shape = Measure)) +

  geom\_line(aes(linetype = Measure)) +

  scale\_x\_continuous(breaks = 1:15 , labels = 1:15) +

  xlab("Number of Clusters") + ylab("Score")

Usando o pacote NbClust

library(NbClust)

set.seed(123)

res.nb <- NbClust(dist.om1, distance = "euclidean",

min.nc = 2, max.nc = 10,

method = "complete", index ="gap")

res.nb

#o pacote nbclust não funciona tão bem, na função all, só na gap. O pacote nbclust sugeriu 2 clusters

cl1.2 <- cutree(clusterward1, k = 2)

cl1.3 <- cutree(clusterward1, k = 3)

cl1.4 <- cutree(clusterward1, k = 4)

seqdplot(seqreg.seq, group = cl1.4, with.legend=T, main = "Estados segundo tipo de trajetória")

seqfplot(seqreg.seq, group = cl1.4, with.legend=T, main = "Seq. mais comuns por grupo")

#Merging the cluster vector with the database

classcluster <- matrix(c(seqreg$cpf,cl1.4),ncol=2)

colnames(classcluster) <- c("cpf","cluster")

seqregclass <- merge(seqreg,classcluster,by="cpf")

attach(seqregclass)

seqregclass$clustername[seqregclass$cluster == 1] <- "serv-priv"

seqregclass$clustername[seqregclass$cluster == 2] <- "serv-pol "

seqregclass$clustername[seqregclass$cluster == 3] <- "servpol-priv "

seqregclass$clustername[seqregclass$cluster == 4] <- "serv"

detach(seqregclass)

#mosaicplot

library(vcd)

#Tipos e agências

ctagencia.labels <- c("agencia", "tipos de trajetória")

ctagencia <- table(seqregclass$Agencia, seqregclass$clustername, dnn = (ctagencia.labels))

mosaic(ctagencia, shade=TRUE, legend=TRUE)

chisagencia <- chisq.test(ctagencia)

chisagencia$stdres

chisagencia$residuals

#Setor privado e politização

ctprivpol.labels <- c("politico", "depois")

ctprivpol <- table(seqregclass$politico, seqregclass$depois, dnn = (ctprivpol.labels))

mosaic(ctprivpol, shade=TRUE, legend=TRUE)

chisprivpol <- chisq.test(ctprivpol)

chisprivpol$stdres

chisprivpol$residuals

#Setor privado e público

ctprivpub.labels <- c("antes", "depois")

ctprivpub <- table(seqregclass$antes, seqregclass$depois, dnn = (ctprivpub.labels))

mosaic(ctprivpub, shade=TRUE, legend=TRUE)

chisprivpub <- chisq.test(ctprivpub)

chisprivpub$stdres

chisprivpub$residuals

#Modelo probit

myprobit <- glm(depois ~ antes + politico, family = binomial(link = "probit"),

data = seqregclass)

summary(myprobit)

#modelo restrito aos funças

seqregclass.pub <- subset(seqregclass, antes=="pub")

myprobit2 <- glm(depois ~ politico, family = binomial(link = "probit"),

data = seqregclass.pub)

summary(myprobit2)