SpotifyProfiling

2023-01-05

## Profiling

# READING CREDSCO\_BIN  
# load("d:/karina/docencia/DataMiningEI/Practiques/2CredscoProfiling/credscok\_bin")  
  
#read data only if required  
# setwd("D:/karina/docencia/01areferenciesPPT/0DadesPractiques/CREDSCO")  
# setwd("C:/Users/Joseph/Documents/Codes/2022/mvtec-2022/mvtec-statsprogramming/statsprog-09")  
# dd <- read.table("data/CredScoClean.csv",header=T, sep=";", dec='.');  
dd <- read.table("data/SpotifyClustersData.csv",header=T, sep=",", dec='.');  
  
names(dd)

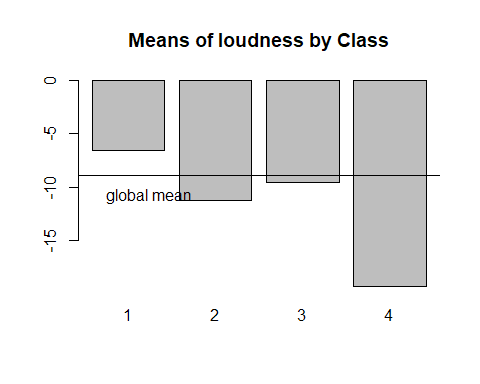
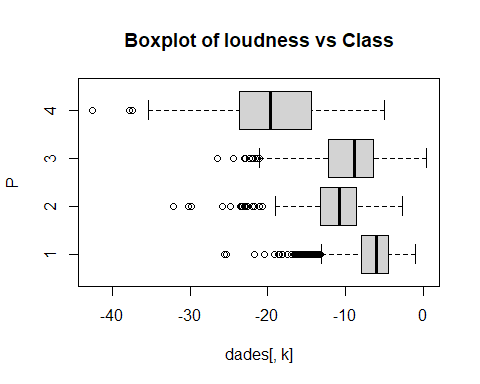
## [1] "loudness" "energy" "acousticness" "instrumentalness"  
## [5] "valence"

attach(dd)  
  
#Dictamen <- as.factor(Dictamen)  
#levels(Dictamen) <- c(NA, "positiu","negatiu")  
  
#Calcula els valor test de la variable Xnum per totes les modalitats del factor P  
ValorTestXnum <- function(Xnum,P){  
 #freq dis of fac  
 nk <- as.vector(table(P));   
 n <- sum(nk);   
 #mitjanes x grups  
 xk <- tapply(Xnum,P,mean);  
 #valors test  
 txk <- (xk-mean(Xnum))/(sd(Xnum)\*sqrt((n-nk)/(n\*nk)));   
 #p-values  
 pxk <- pt(txk,n-1,lower.tail=F);  
 for(c in 1:length(levels(as.factor(P)))){if (pxk[c]>0.5){pxk[c]<-1-pxk[c]}}  
 return (pxk)  
}  
  
  
ValorTestXquali <- function(P,Xquali){  
 taula <- table(P,Xquali);  
 n <- sum(taula);   
 pk <- apply(taula,1,sum)/n;  
 pj <- apply(taula,2,sum)/n;  
 pf <- taula/(n\*pk);  
 pjm <- matrix(data=pj,nrow=dim(pf)[1],ncol=dim(pf)[2], byrow=TRUE);   
 dpf <- pf - pjm;   
 dvt <- sqrt(((1-pk)/(n\*pk))%\*%t(pj\*(1-pj)));   
 #i hi ha divisions iguals a 0 dona NA i no funciona  
 zkj <- dpf  
 zkj[dpf!=0]<-dpf[dpf!=0]/dvt[dpf!=0];   
 pzkj <- pnorm(zkj,lower.tail=F);  
 for(c in 1:length(levels(as.factor(P)))){for (s in 1:length(levels(Xquali))){if (pzkj[c,s]> 0.5){pzkj[c,s]<-1- pzkj[c,s]}}}  
 return (list(rowpf=pf,vtest=zkj,pval=pzkj))  
}  
  
  
#source("file")  
#dades contain the dataset  
dades<-dd  
#dades<-dd[filtro,]  
#dades<-df  
K<-dim(dades)[2]  
par(ask=TRUE)  
  
#--- Inserted a code from generated c2 variable from clustering  
## Dissimilarity matrix (Main task)  
 library(cluster)  
 numeric\_vars <- c("loudness", "energy", "acousticness", "instrumentalness", "valence")  
 dissimMatrix <- daisy(dd[,numeric\_vars], metric = "gower", stand=TRUE)   
 distMatrix<-dissimMatrix^2  
   
 ## INSTRUCTION THAT RUN THE HEIRARCHICAL CLUSTERING  
 h1 <- hclust(distMatrix,method="ward.D") # NOTICE THE COST  
   
 ## CUT THE TREE  
 c2 <- cutree(h1, 4)  
#---  
  
#P must contain the class variable  
#P<-dd[,3]  
P<-c2  
#P<-dd[,18]  
nameP<-"classe"  
#P<-df[,33]  
  
nc<-length(levels(factor(P)))  
nc

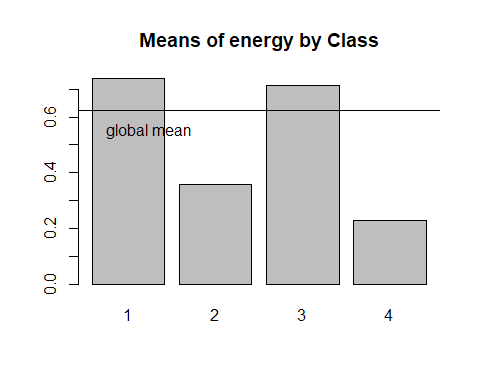
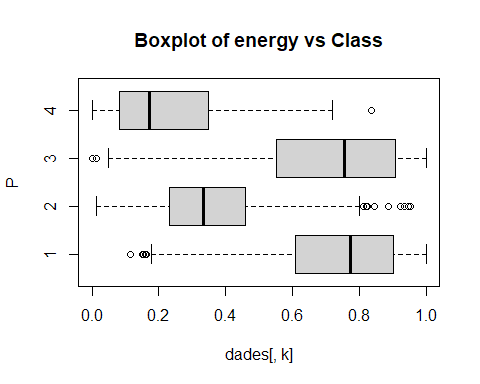
## [1] 4

pvalk <- matrix(data=0,nrow=nc,ncol=K, dimnames=list(levels(P),names(dades)))  
nameP<-"Class"  
n<-dim(dades)[1]  
  
for(k in 1:K){  
 if (is.numeric(dades[,k])){   
 print(paste("Anàlisi per classes de la Variable:", names(dades)[k]))  
   
 boxplot(dades[,k]~P, main=paste("Boxplot of", names(dades)[k], "vs", nameP ), horizontal=TRUE)  
   
 barplot(tapply(dades[[k]], P, mean),main=paste("Means of", names(dades)[k], "by", nameP ))  
 abline(h=mean(dades[[k]]))  
 legend(0,mean(dades[[k]]),"global mean",bty="n")  
 print("Estadístics per groups:")  
 for(s in levels(as.factor(P))) {print(summary(dades[P==s,k]))}  
 o<-oneway.test(dades[,k]~P)  
 print(paste("p-valueANOVA:", o$p.value))  
 kw<-kruskal.test(dades[,k]~P)  
 print(paste("p-value Kruskal-Wallis:", kw$p.value))  
 pvalk[,k]<-ValorTestXnum(dades[,k], P)  
 print("p-values ValorsTest: ")  
 print(pvalk[,k])   
 }else{  
 if(class(dd[,k])=="Date"){  
 print(summary(dd[,k]))  
 print(sd(dd[,k]))  
 #decide breaks: weeks, months, quarters...  
 hist(dd[,k],breaks="weeks")  
 }else{  
 #qualitatives  
 print(paste("Variable", names(dades)[k]))  
 table<-table(P,dades[,k])  
 # print("Cross-table")  
 # print(table)  
 rowperc<-prop.table(table,1)  
   
 colperc<-prop.table(table,2)  
 # print("Distribucions condicionades a files")  
 # print(rowperc)  
   
 #ojo porque si la variable es true o false la identifica amb el tipus Logical i  
 #aquest no te levels, por tanto, coertion preventiva  
   
 dades[,k]<-as.factor(dades[,k])  
   
   
 marg <- table(as.factor(P))/n  
 print(append("Categories=",levels(as.factor(dades[,k]))))  
   
 #from next plots, select one of them according to your practical case  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }  
   
 #with legend  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }  
 legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)  
   
 #condicionades a classes  
 print(append("Categories=",levels(dades[,k])))  
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }  
   
 #with legend  
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]))  
 paleta<-rainbow(length(levels(dades[,k])))  
 for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }  
 legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)  
   
 #amb variable en eix d'abcisses  
 marg <-table(dades[,k])/n  
 print(append("Categories=",levels(dades[,k])))  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 #x<-plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), xaxt="n")  
 #text(x=x+.25, y=-1, adj=1, levels(CountryName), xpd=TRUE, srt=25, cex=0.7)  
 paleta<-rainbow(length(levels(as.factor(P))))  
 for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c]) }  
   
 #with legend  
 plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c])}  
 legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)  
   
 #condicionades a columna   
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 paleta<-rainbow(length(levels(as.factor(P))))  
 for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c]) }  
   
 #with legend  
 plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of pos & neg by",names(dades)[k]), las=3)  
 for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c])}  
 legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)  
   
 table<-table(dades[,k],P)  
 print("Cross Table:")  
 print(table)  
 print("Distribucions condicionades a columnes:")  
 print(colperc)  
   
 #diagrames de barres apilades   
   
 paleta<-rainbow(length(levels(dades[,k])))  
 barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )  
   
 barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )  
 legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)  
   
 #diagrames de barres adosades  
 barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta )  
   
 barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta)  
 legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)  
   
 print("Test Chi quadrat: ")  
 print(chisq.test(dades[,k], as.factor(P)))  
   
 print("valorsTest:")  
 print( ValorTestXquali(P,dades[,k]))  
 #calcular els pvalues de les quali  
 }  
 }  
}#endfor

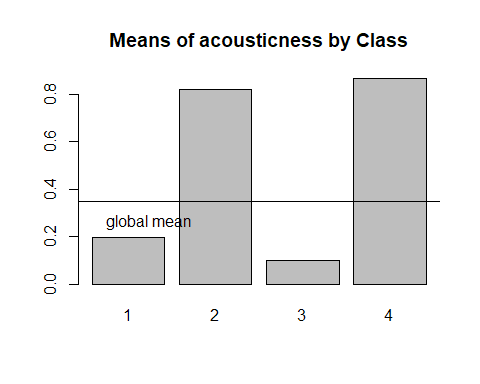
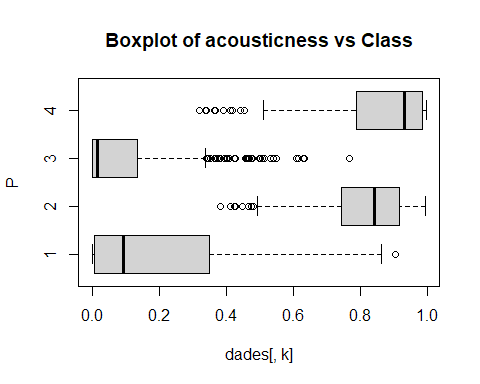
## [1] "Anàlisi per classes de la Variable: loudness"



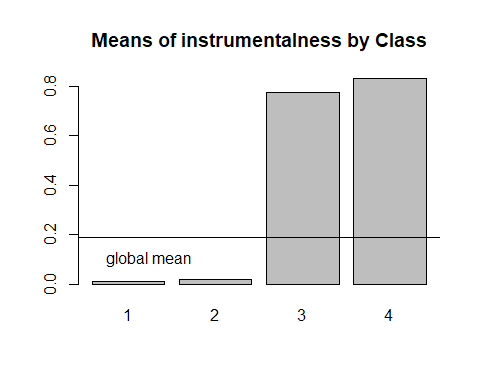
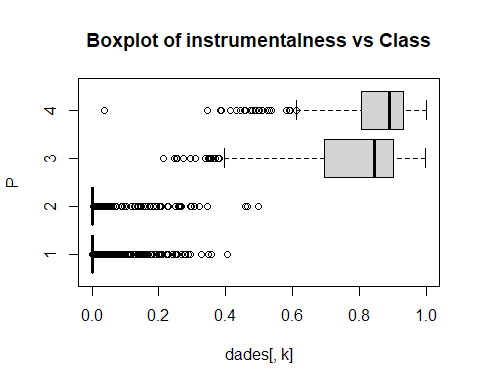
## [1] "Estadístics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -25.641 -7.987 -6.035 -6.583 -4.519 -0.958   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -32.108 -13.261 -10.819 -11.236 -8.603 -2.689   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -26.527 -12.243 -8.899 -9.548 -6.371 0.377   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## -42.631 -23.609 -19.674 -19.321 -14.454 -5.044   
## [1] "p-valueANOVA: 2.17489695592098e-173"  
## [1] "p-value Kruskal-Wallis: 2.45213703529884e-239"  
## [1] "p-values ValorsTest: "  
## [1] 3.451881e-166 0.000000e+00 3.369412e-03 0.000000e+00  
## [1] "Anàlisi per classes de la Variable: energy"



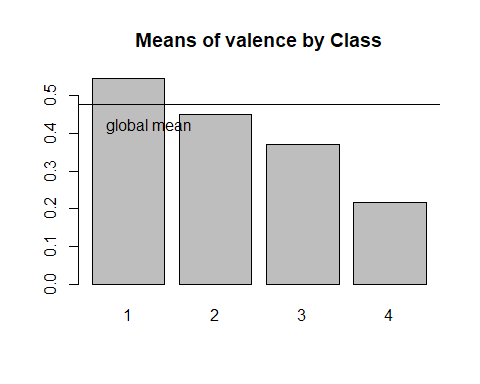
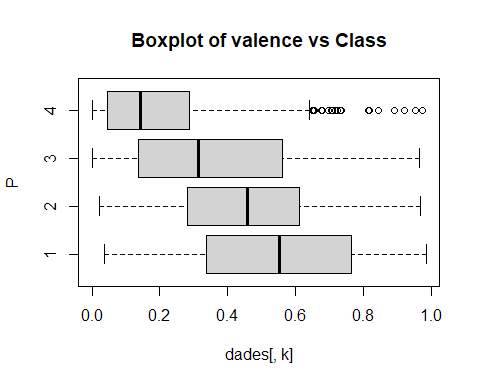
## [1] "Estadístics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.1140 0.6070 0.7710 0.7381 0.9000 0.9990   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0125 0.2315 0.3325 0.3562 0.4590 0.9530   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000242 0.551500 0.755500 0.714677 0.906000 0.999000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00108 0.08330 0.17200 0.22836 0.34800 0.83600   
## [1] "p-valueANOVA: 1.24124380786274e-274"  
## [1] "p-value Kruskal-Wallis: 1.11360835084654e-278"  
## [1] "p-values ValorsTest: "  
## [1] 5.555962e-161 0.000000e+00 1.102107e-13 0.000000e+00  
## [1] "Anàlisi per classes de la Variable: acousticness"



## [1] "Estadístics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000001 0.006440 0.093600 0.197365 0.348500 0.905000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3830 0.7438 0.8410 0.8189 0.9170 0.9950   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000011 0.0003835 0.0147000 0.0970514 0.1352500 0.7660000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.3190 0.7870 0.9310 0.8646 0.9850 0.9960   
## [1] "p-valueANOVA: 0"  
## [1] "p-value Kruskal-Wallis: 0"  
## [1] "p-values ValorsTest: "  
## [1] 0.000000e+00 6.393416e-212 0.000000e+00 2.450066e-123  
## [1] "Anàlisi per classes de la Variable: instrumentalness"



## [1] "Estadístics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000000 0.0000000 0.0000049 0.0121457 0.0006955 0.4060000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000000 0.000000 0.000006 0.019713 0.001240 0.498000   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.2130 0.6940 0.8440 0.7773 0.9010 0.9990   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0379 0.8050 0.8900 0.8312 0.9330 1.0000   
## [1] "p-valueANOVA: 0"  
## [1] "p-value Kruskal-Wallis: 0"  
## [1] "p-values ValorsTest: "  
## [1] 0.000000e+00 0.000000e+00 1.726791e-258 6.493344e-192  
## [1] "Anàlisi per classes de la Variable: valence"



## [1] "Estadístics per groups:"  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0366 0.3370 0.5530 0.5458 0.7635 0.9850   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0219 0.2830 0.4585 0.4514 0.6085 0.9680   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.1378 0.3130 0.3702 0.5613 0.9640   
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0454 0.1430 0.2167 0.2860 0.9730   
## [1] "p-valueANOVA: 6.3023780909863e-92"  
## [1] "p-value Kruskal-Wallis: 4.24544276547456e-94"  
## [1] "p-values ValorsTest: "  
## [1] 7.842021e-65 7.847820e-03 0.000000e+00 0.000000e+00

#descriptors de les classes més significatius. Afegir info qualits  
for (c in 1:length(levels(as.factor(P)))) {  
 if(!is.na(levels(as.factor(P))[c])){  
 print(paste("P.values per class:",levels(as.factor(P))[c]));  
 print(sort(pvalk[c,]), digits=3)   
 }  
}

## [1] "P.values per class: 1"  
## acousticness instrumentalness loudness energy   
## 0.00e+00 0.00e+00 3.45e-166 5.56e-161   
## valence   
## 7.84e-65   
## [1] "P.values per class: 2"  
## loudness energy instrumentalness acousticness   
## 0.00e+00 0.00e+00 0.00e+00 6.39e-212   
## valence   
## 7.85e-03   
## [1] "P.values per class: 3"  
## acousticness valence instrumentalness energy   
## 0.00e+00 0.00e+00 1.73e-258 1.10e-13   
## loudness   
## 3.37e-03   
## [1] "P.values per class: 4"  
## loudness energy valence instrumentalness   
## 0.00e+00 0.00e+00 0.00e+00 6.49e-192   
## acousticness   
## 2.45e-123

#afegir la informacio de les modalitats de les qualitatives a la llista de pvalues i fer ordenacio global  
  
#saving the dataframe in an external file  
#write.table(dd, file = "credscoClean.csv", sep = ";", na = "NA", dec = ".", row.names = FALSE, col.names = TRUE)