Hierarchical Cluestering 27-11

Adrià Casanova

2023-11-27

# Clear plots  
if(!is.null(dev.list())) dev.off()

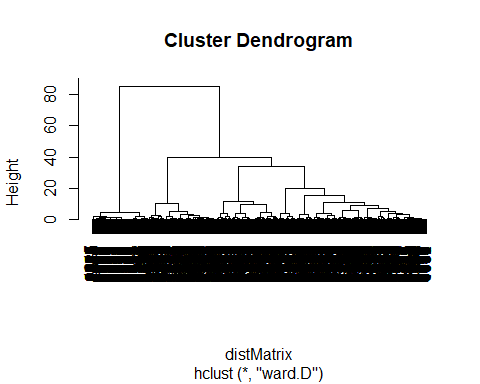
## null device   
## 1

# Clean workspace  
rm(list=ls())  
  
# Import library  
library(cluster)

# Import data  
dd <- read.csv("2.Bank\_India\_preprocessed\_data.csv", sep=",",  
 stringsAsFactors=TRUE)  
# remove the id  
dd$id <- NULL  
class(dd)

## [1] "data.frame"

# attach(dd)  
  
actives<-c(1:19) # We have already removed the id variables  
# gower distance  
dissimMatrix <- daisy(dd[,actives], metric = "gower", stand=TRUE)  
  
distMatrix<-dissimMatrix^2  
  
set.seed(123)  
h1 <- hclust(distMatrix,method="ward.D") # NOTICE THE COST  
  
# Plot the dendogram  
plot(h1)



# Cut the tree with cluster equal to 4  
c2 <- cutree(h1,4)

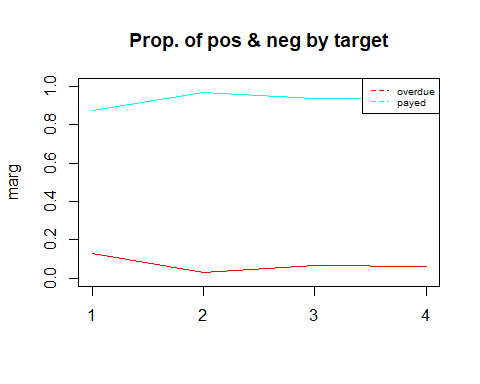
#Calcula els valor test de la variable Xnum per totes les modalitats del factor P  
ValorTestXnum <- function(Xnum,P){  
 #freq dis of fac  
 nk <- as.vector(table(P));   
 n <- sum(nk);   
 #mitjanes x grups  
 xk <- tapply(Xnum,P,mean);  
 #valors test  
 txk <- (xk-mean(Xnum))/(sd(Xnum)\*sqrt((n-nk)/(n\*nk)));   
 #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))  
}  
  
dades<-dd  
  
K<-dim(dades)[2]  
  
  
#P must contain the class variable  
#P<-dd[,3]  
P<-c2 #P CONTAINS THE NUMBER OF CUTS WE DO ON THE TREE  
  
nc <-length(levels(factor(P)))  
nc

## [1] 4

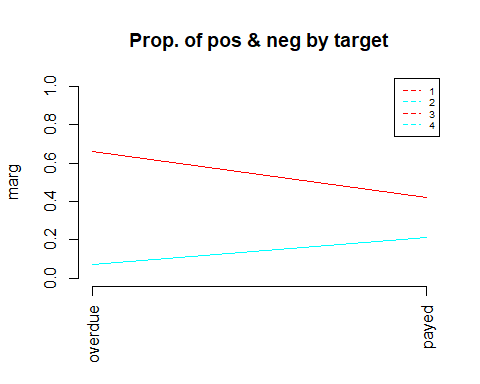
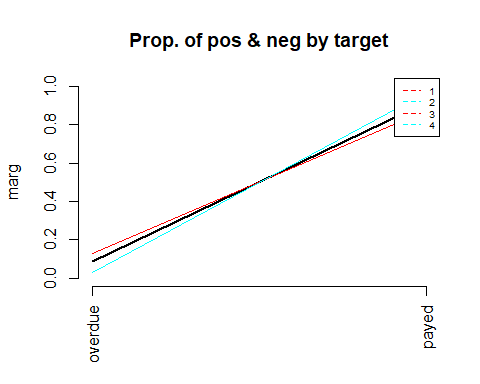
pvalk <- matrix(data=0,nrow=nc,ncol=K, dimnames=list(levels(P),names(dades)))  
nameP<-"Class"  
n <- dim(dades)[1]

for(k in 1:4){  
 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

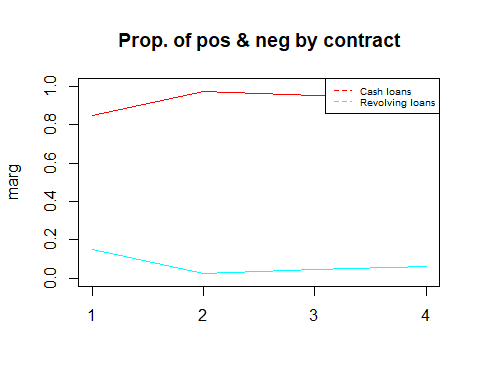
## [1] "Variable target"  
## [1] "Categories=" "overdue" "payed"   
## [1] "Categories=" "overdue" "payed"



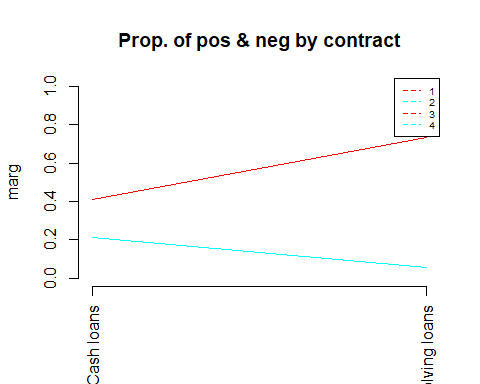
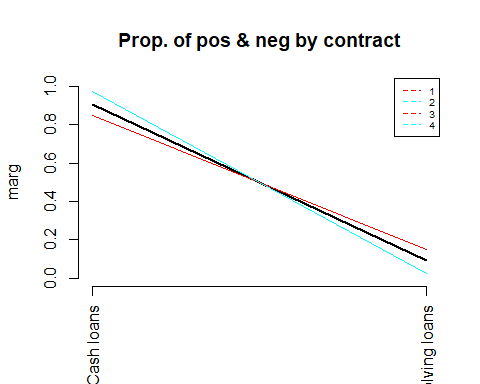
## [1] "Categories=" "overdue" "payed"



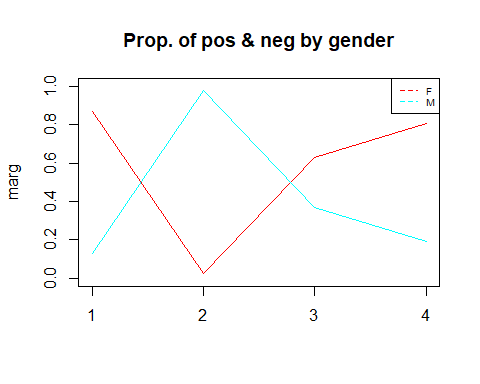
## [1] "Cross Table:"  
## P  
## 1 2 3 4  
## overdue 282 30 61 54  
## payed 1924 967 877 804  
## [1] "Distribucions condicionades a columnes:"  
##   
## P overdue payed  
## 1 0.66042155 0.42082240  
## 2 0.07025761 0.21150481  
## 3 0.14285714 0.19181977  
## 4 0.12646370 0.17585302  
## [1] "Test Chi quadrat: "  
##   
## Pearson's Chi-squared test  
##   
## data: dades[, k] and as.factor(P)  
## X-squared = 100.41, df = 3, p-value < 2.2e-16  
##   
## [1] "valorsTest:"  
## $rowpf  
## Xquali  
## P overdue payed  
## 1 0.12783318 0.87216682  
## 2 0.03009027 0.96990973  
## 3 0.06503198 0.93496802  
## 4 0.06293706 0.93706294  
##   
## $vtest  
## Xquali  
## P overdue payed  
## 1 9.535768 -9.535768  
## 2 -6.985578 6.985578  
## 3 -2.478309 2.478309  
## 4 -2.588483 2.588483  
##   
## $pval  
## Xquali  
## P overdue payed  
## 1 7.439028e-22 0.000000e+00  
## 2 1.418421e-12 1.418428e-12  
## 3 6.600340e-03 6.600340e-03  
## 4 4.819979e-03 4.819979e-03  
##   
## [1] "Variable contract"  
## [1] "Categories=" "Cash loans" "Revolving loans"  
## [1] "Categories=" "Cash loans" "Revolving loans"



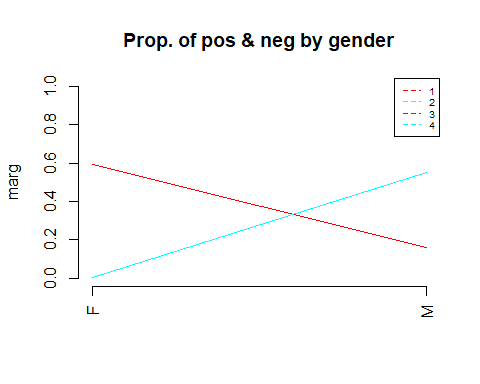
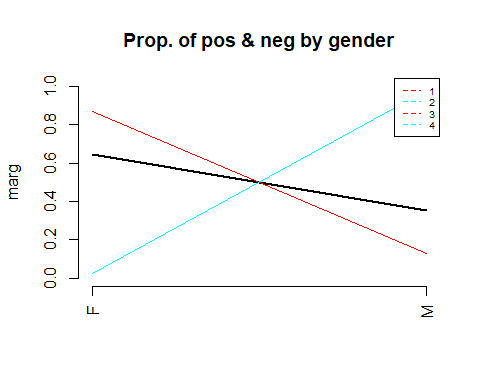
## [1] "Categories=" "Cash loans" "Revolving loans"



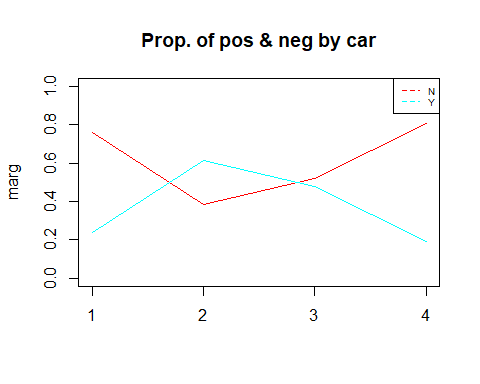
## [1] "Cross Table:"  
## P  
## 1 2 3 4  
## Cash loans 1869 972 894 804  
## Revolving loans 337 25 44 54  
## [1] "Distribucions condicionades a columnes:"  
##   
## P Cash loans Revolving loans  
## 1 0.41176471 0.73260870  
## 2 0.21414408 0.05434783  
## 3 0.19695968 0.09565217  
## 4 0.17713153 0.11739130  
## [1] "Test Chi quadrat: "  
##   
## Pearson's Chi-squared test  
##   
## data: dades[, k] and as.factor(P)  
## X-squared = 182.44, df = 3, p-value < 2.2e-16  
##   
## [1] "valorsTest:"  
## $rowpf  
## Xquali  
## P Cash loans Revolving loans  
## 1 0.84723481 0.15276519  
## 2 0.97492477 0.02507523  
## 3 0.95309168 0.04690832  
## 4 0.93706294 0.06293706  
##   
## $vtest  
## Xquali  
## P Cash loans Revolving loans  
## 1 -13.205543 13.205543  
## 2 8.172992 -8.172992  
## 3 5.303033 -5.303033  
## 4 3.237958 -3.237958  
##   
## $pval  
## Xquali  
## P Cash loans Revolving loans  
## 1 0.000000e+00 4.075494e-40  
## 2 1.504173e-16 1.110223e-16  
## 3 5.694728e-08 5.694728e-08  
## 4 6.019436e-04 6.019436e-04  
##   
## [1] "Variable gender"  
## [1] "Categories=" "F" "M"   
## [1] "Categories=" "F" "M"



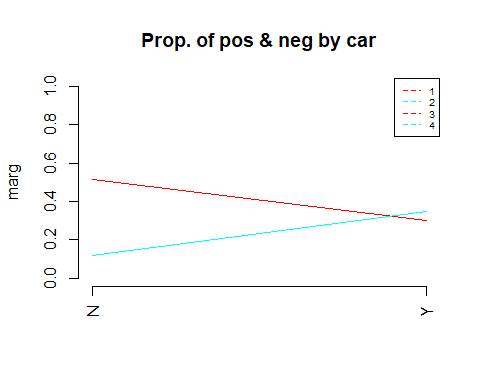
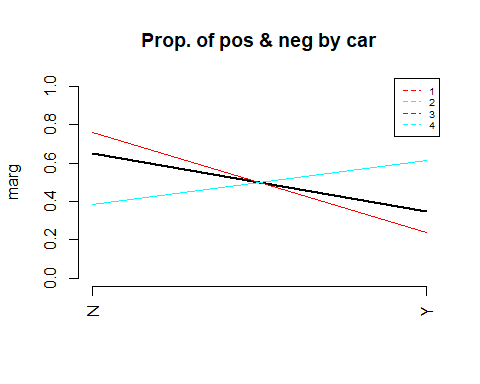
## [1] "Categories=" "F" "M"



## [1] "Cross Table:"  
## P  
## 1 2 3 4  
## F 1921 23 590 693  
## M 285 974 348 165  
## [1] "Distribucions condicionades a columnes:"  
##   
## P F M  
## 1 0.595289743 0.160835214  
## 2 0.007127363 0.549661400  
## 3 0.182832352 0.196388262  
## 4 0.214750542 0.093115124  
## [1] "Test Chi quadrat: "  
##   
## Pearson's Chi-squared test  
##   
## data: dades[, k] and as.factor(P)  
## X-squared = 2277.2, df = 3, p-value < 2.2e-16  
##   
## [1] "valorsTest:"  
## $rowpf  
## Xquali  
## P F M  
## 1 0.87080689 0.12919311  
## 2 0.02306921 0.97693079  
## 3 0.62899787 0.37100213  
## 4 0.80769231 0.19230769  
##   
## $vtest  
## Xquali  
## P F M  
## 1 29.592320 -29.592320  
## 2 -45.921258 45.921258  
## 3 -1.174312 1.174312  
## 4 10.910300 -10.910300  
##   
## $pval  
## Xquali  
## P F M  
## 1 9.380435e-193 0.000000e+00  
## 2 0.000000e+00 0.000000e+00  
## 3 1.201349e-01 1.201349e-01  
## 4 5.145793e-28 0.000000e+00  
##   
## [1] "Variable car"  
## [1] "Categories=" "N" "Y"   
## [1] "Categories=" "N" "Y"



## [1] "Categories=" "N" "Y"



## [1] "Cross Table:"  
## P  
## 1 2 3 4  
## N 1679 386 489 691  
## Y 527 611 449 167  
## [1] "Distribucions condicionades a columnes:"  
##   
## P N Y  
## 1 0.51741140 0.30045610  
## 2 0.11895223 0.34834664  
## 3 0.15069337 0.25598632  
## 4 0.21294299 0.09521095  
## [1] "Test Chi quadrat: "  
##   
## Pearson's Chi-squared test  
##   
## data: dades[, k] and as.factor(P)  
## X-squared = 581.08, df = 3, p-value < 2.2e-16  
##   
## [1] "valorsTest:"  
## $rowpf  
## Xquali  
## P N Y  
## 1 0.7611061 0.2388939  
## 2 0.3871615 0.6128385  
## 3 0.5213220 0.4786780  
## 4 0.8053613 0.1946387  
##   
## $vtest  
## Xquali  
## P N Y  
## 1 14.743336 -14.743336  
## 2 -19.371373 19.371373  
## 3 -9.100077 9.100077  
## 4 10.535669 -10.535669  
##   
## $pval  
## Xquali  
## P N Y  
## 1 1.697939e-49 0.000000e+00  
## 2 0.000000e+00 6.730949e-84  
## 3 0.000000e+00 4.513395e-20  
## 4 2.958043e-26 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"  
## target contract gender car n\_child income   
## 0 0 0 0 0 0   
## credit loan price job\_stat studies family   
## 0 0 0 0 0 0   
## house age job\_duration occupation job\_type n\_enquiries   
## 0 0 0 0 0 0   
## companion   
## 0   
## [1] "P.values per class: 2"  
## target contract gender car n\_child income   
## 0 0 0 0 0 0   
## credit loan price job\_stat studies family   
## 0 0 0 0 0 0   
## house age job\_duration occupation job\_type n\_enquiries   
## 0 0 0 0 0 0   
## companion   
## 0   
## [1] "P.values per class: 3"  
## target contract gender car n\_child income   
## 0 0 0 0 0 0   
## credit loan price job\_stat studies family   
## 0 0 0 0 0 0   
## house age job\_duration occupation job\_type n\_enquiries   
## 0 0 0 0 0 0   
## companion   
## 0   
## [1] "P.values per class: 4"  
## target contract gender car n\_child income   
## 0 0 0 0 0 0   
## credit loan price job\_stat studies family   
## 0 0 0 0 0 0   
## house age job\_duration occupation job\_type n\_enquiries   
## 0 0 0 0 0 0   
## companion   
## 0

#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)