```r
library(lattice)
library(forcats)
library(DescTools)
library(Factoshiny)
library(FactoMineR)
#modifier le chemin
heartt<- read.csv(file = '/Users/baganouiemna/Desktop/heart.csv',header = TRUE,
                  stringsAsFactors = TRUE)
heart_sub <- subset(heartt,HeartDisease =="Yes")
set.seed(42)
rand_heart <- heart_sub[sample(nrow(heart_sub), size=500), ]
rand_heart <-select(rand_heart,-HeartDisease)

describe(rand_heart)
summary(rand_heart)


quali <- which(sapply(rand_heart, is.factor))
quanti <- which(sapply(rand_heart, is.numeric))
quanti.discr <- which(sapply(rand_heart,
                             FUN = function(xx){
                               (is.numeric(xx))&(length(table(xx)))<30
                             })
)
quanti.cont <- quanti[-quanti.discr]
# --------------------------------affichages
graphiques------------------------------------
## choix du nombre de classes par la regle de Sturges
n <- nrow(rand_heart)#nombre d'individus
k <- ceiling(1 + log(n)/log(2))#nombre de classes
## construction de l'histogramme
histogram(rand_heart$BMI, nint = k, type = "count")

## construction de l'histogramme
histogram(rand_heart$BMI, nint = k, type = "count")
##test de normalite
shapiro.test(rand_heart$BMI)

mapply(rand_heart[,quanti],
       FUN = function(xx,name){boxplot(xx, main = name,col = "slategray")},
       name = names(quanti))

### Diagrammes en barres
par(mfrow = c(2, 3))
mapply(rand_heart[,quanti.discr],
       FUN = function(xx,name){barplot(table(xx),main = name)},
       name = names(quanti.discr))

mapply(rand_heart[,quali],
       FUN = function(xx,name){barplot(table(xx),main = name)},
       name = names(quali))

##----------------------------discretisation|regroupement de
modalites---------------------------
rand_heart$Diabetic <-as.character(rand_heart$Diabetic)
rand_heart$Diabetic[rand_heart$Diabetic %in%  c("Yes","Yes (during pregnancy)")]
<- "Yes"
rand_heart$Diabetic[rand_heart$Diabetic %in%  c("No","No, borderline diabetes")]
<- "No"
rand_heart$Diabetic<- factor(rand_heart$Diabetic)
table(rand_heart$Diabetic)

rand_heart$Race<-as.character(rand_heart$Race)
rand_heart$Race[rand_heart$Race %in%  c("White")] <- "White"
```

```r
rand_heart$Race[rand_heart$Race %in%  c("Hispanic","Other", "borderline
diabetes","American Indian/Alaskan Native","Asian","Black","")] <- "Other"
rand_heart$Race <- factor(rand_heart$Race)
table(rand_heart$Race)

rand_heart$AgeCategory <-as.character(rand_heart$AgeCategory)
rand_heart$AgeCategory[rand_heart$AgeCategory %in%  c("18-24", "25-29","30-
34","35-39","40-44","45-49")] <- "moins de 50 ans"
rand_heart$AgeCategory[rand_heart$AgeCategory %in%  c("50-54","55-59","60-64")]
<- "50-65"
rand_heart$AgeCategory[rand_heart$AgeCategory %in%  c("65-69","70-74", "75-79",
"80 or older")] <- "65 or more"
rand_heart$AgeCategory <- fct_relevel(
        rand_heart$AgeCategory,"moins de 50 ans","50-65","65 or more"

)
rand_heart$AgeCategory <- factor(rand_heart$AgeCategory)
table(rand_heart$AgeCategory)

rand_heart$MentalHealth<-as.character(rand_heart$MentalHealth)
rand_heart$MentalHealth[rand_heart$MentalHealth %in%  c("0")] <- "Aucun jour"
rand_heart$MentalHealth[rand_heart$MentalHealth %in%
c("1","2","3","4","5","7" ,"8", "10", "12","14","15","20" ,"23","25")] <-
"Nombreux jours"
rand_heart$MentalHealth[rand_heart$MentalHealth %in%  c("30")] <- "Un mois"
rand_heart$MentalHealth <- factor(rand_heart$MentalHealth)
table(rand_heart$MentalHealth)

rand_heart$PhysicalHealth <-as.character(rand_heart$PhysicalHealth)
rand_heart$PhysicalHealth[rand_heart$PhysicalHealth %in%  c("0")] <- "Aucun
jour"
rand_heart$PhysicalHealth[rand_heart$PhysicalHealth %in%
c("1","2","3","4","5","6","9","7" ,"8", "10", "12","14","15","17","19","20","21"
,"28","25")] <- "Nombreux jours"
rand_heart$PhysicalHealth[rand_heart$PhysicalHealth %in%  c("30")] <- "Un mois"
rand_heart$PhysicalHealth <- factor(rand_heart$PhysicalHealth)
table(rand_heart$PhysicalHealth)

BreakBMI = c(15.5, 18.5, 25,30, max(rand_heart$BMI))
rand_heart$BMI= cut(rand_heart$BMI, breaks = BreakBMI, include.lowest =
TRUE,labels = c("Sous-poid", "poid-Normal", "Surpoids", "Obésité/obésité
morbide"))
summary(rand_heart$BMI)
table(rand_heart$BMI)
freq(rand_heart$BMI)
barplot(table(rand_heart$BMI))

BreakSleepTime = c(2, 6, 9,max(rand_heart$SleepTime))
rand_heart$SleepTime = cut(rand_heart$SleepTime, breaks = BreakSleepTime,
include.lowest = TRUE,labels = c("2h Ã  5h", "6h Ã  8h", "9h et plus"))
summary(rand_heart$SleepTime)
table(SleepTime)
freq(SleepTime)
barplot(table(SleepTime))

#-------------------calcul des
effectifs---------------------------------------------------------
#REMPLACE LA VARIABLE
frequence <- freq(rand_heart$Stroke)
frequence
barplot(frequence$`%`,
        ylab = "fréquence relative",
        names.arg = rownames(frequence))
```

```
#--------------------------------Tests statistiques et tableaux profils-
lignes--------------------------
# On ecrit pas toutes les combinaisons de variables (on remplace)
crossTabDepRef<- table(rand_heart$AgeCategory,rand_heart$Asthma)
crossTabDepRef
crossTabDepRef<- table(rand_heart$AgeCategory,rand_heart$SkinCancer)
crossTabDepRef
fishertest <- fisher.test(table(rand_heart$AgeCategory,
rand_heart$SkinCancer),workspace = 2e8)
fishertest
testdukhideux <- chisq.test(table(rand_heart$AgeCategory, rand_heart$Asthma))
testdukhideux
crossTabDepRef<- table(rand_heart$AgeCategory,rand_heart$BMI)
crossTabDepRef
testdukhideux <- chisq.test(table(rand_heart$AgeCategory, rand_heart$Diabetic))
testdukhideux
fishertest <- fisher.test(table(rand_heart$AgeCategory,
rand_heart$KidneyDisease),workspace = 2e8)
fishertest
testdukhideux <- chisq.test(table(rand_heart$AgeCategory, rand_heart$Stroke))
testdukhideux
fishertest <- fisher.test(table(rand_heart$AgeCategory,
rand_heart$BMI),workspace = 2e8)
fishertest
crossTabDepRef<- table(rand_heart$Race,rand_heart$Asthma)
crossTabDepRef
crossTabDepRef<- table(rand_heart$Race,rand_heart$KidneyDisease)
crossTabDepRef
testdukhideux <- chisq.test(table(rand_heart$Race, rand_heart$KidneyDisease))
testdukhideux
crossTabDepRef<- table(rand_heart$Race,rand_heart$Diabetic)
crossTabDepRef
testdukhideux <- chisq.test(table(rand_heart$Race, rand_heart$Diabetic))
testdukhideux
testdukhideux <- chisq.test(table(rand_heart$Race, rand_heart$Stroke))
testdukhideux
fishertest <- fisher.test(table(rand_heart$Race, rand_heart$BMI),workspace =
2e8)
fishertest
crossTabDepRef<- table(rand_heart$Sex,rand_heart$Asthma)
crossTabDepRef
testdukhideux <- chisq.test(table(rand_heart$Sex,rand_heart$KidneyDisease))
testdukhideux
fishertest <- fisher.test(table(rand_heart$Sex,rand_heart$BMI),workspace = 2e8)
fishertest
#------------
cramer-----------------------------------------------------------------------
-------

cramer.v(table(rand_heart$AgeCategory, rand_heart$BMI))
cramer.v(table(rand_heart$Race, rand_heart$KidneyDisease))
cramer.v(table(rand_heart$Sex, rand_heart$Asthma))
#---------------profils-
lignes----------------------------------------------------------------
#(on change le nom de la variable si besoin)
lprop(table(rand_heart$AgeCategory,rand_heart$SkinCancer))
lprop(table(rand_heart$Race, rand_heart$KidneyDisease))
lprop(table(rand_heart$Sex,rand_heart$Asthma))

#-----------ACM (essentiellement avec
Factoshiny)--------------------------------------------------------------
----------------
#apres suppression de quelques inidividus
```

```
New_rand_heart <- subset( rand_heart , rand_heart$BMI != "Sous-poid")

tt<-Factoshiny(New_rand_heart)

#axe 1:2
res1.MCA<-MCA(New_rand_heart,quali.sup=c(8,9,10),graph=FALSE)
plot.MCA(res1.MCA, choix='var',title="Graphe des variables",axes = 1:2)
plot.MCA(res1.MCA,invisible= 'quali.sup',title="Graphe de l'ACM",label
=c('ind','var'))
plot.MCA(res1.MCA,invisible= c('ind','quali.sup'),selectMod= 'contrib  14
',title="Graphe de l'ACM",label =c('var'))
#graphe individus qui contribuent
plot.MCA(res1.MCA,invisible= c('var','quali.sup'),select= 'contrib
139',title="Graphe de l'ACM",label =c('ind'))
#graphe modalites supplementaires
plot.MCA(res1.MCA,invisible= c('ind','var'),title="Graphe de l'ACM",label
=c('quali.sup'))

#axe 3:4
plot.MCA(res1.MCA,axes=c(3,4),invisible= 'quali.sup',title="Graphe de
l'ACM",label =c('ind','var'))
plot.MCA(res1.MCA, choix='var',title="Graphe des variables",axes=c(3,4))
plot.MCA(res1.MCA,axes=c(3,4),invisible= c('ind','quali.sup'),selectMod=
'contrib  14 ',title="Graphe de l'ACM",label =c('var'))
#graphe individus qui contribuent
plot.MCA(res1.MCA,axes=c(3,4),invisible= c('var','quali.sup'),select= 'contrib
139',title="Graphe de l'ACM",label =c('ind'))
#graphe modalites supplementaires
plot.MCA(res1.MCA,axes=c(3,4),invisible= c('ind','var'),title="Graphe de
l'ACM",label =c('quali.sup'))




#-----regle a la kaiser
--------------------------------------------------------------------------------
#------ visualiser
l'inertie---------------------------------------------------------------------
--
eig.val <- res1.MCA$eig
barplot(eig.val[, 2],
        names.arg = 1:nrow(eig.val),
        main = "Inertie expliquee par les dimensions (%)",
        xlab = "Principal Dimensions",
        ylab = "Pourcentage d'inertie",
        col ="steelblue")
# Add connected line segments to the plot
lines(x = 1:nrow(eig.val), eig.val[, 2],
      type = "b", pch = 19, col = "red")

#--visualiser valeur
propres-----------------------------------------------------------------------
res1.MCA$eig
#mean(eig.val[,1]) #0.07142857
eig.val <- res1.MCA$eig
barplot(eig.val[, 1],
        names.arg = 1:nrow(eig.val),
        main = "Les valeures propres par dimension",
        xlab = "Principal Dimensions",
        ylab = "valeur propre",
        col ="steelblue")

#----------------graphe contribution
modalites----------------------------------------------
# Contributions des variables Ã  la dimension 1
```

```
fviz_contrib (res1.MCA, choice = "var", axes = 1, top = 25)
# Contributions des variables à la dimension 2
fviz_contrib (res1.MCA, choice = "var", axes = 2, top = 25)
# Contributions des variables à la dimension 3
fviz_contrib (res1.MCA, choice = "var", axes = 3, top = 25)
# Contributions des variables à la dimension 4
fviz_contrib (res1.MCA, choice = "var", axes = 4, top = 25)

#----------description des
dimensions-------------------------------------------------------------
#------description
automatique--------------------------------------------------------------
-
# Description de la dimension 1
res.desc <- dimdesc (res1.MCA, axes = c(1))
res.desc[[1]]
# Description de la dimension 2
res.desc <- dimdesc (res1.MCA, axes = c(2))
res.desc[[1]]
# Description de la dimension 3
res.desc <- dimdesc (res1.MCA, axes = c(3))
res.desc[[1]]
# Description de la dimension 4
res.desc <- dimdesc (res1.MCA, axes = c(4))
lapply(res.desc[[1]],round,3)

#---------------------description plus
detailler---------------------------------------------
summary(res1.MCA)

res = dimdesc(res1.MCA, axes=4, proba=0.05)
res.var <- res1.MCA$var
res.var$coord
res.var$contrib
res.ind <- res1.MCA$ind
res.ind$coord           # Coordonnees
res.ind$contrib         # Contributions
res.ind$cos2

#---------------------
CAH--------------------------------------------------------------------------
res2.MCA<-MCA(New_rand_heart,ncp=17,quali.sup=c(8,9,10),graph=FALSE)
ncp <-17
D <- dist(res2.MCA$ind$coord[,1:ncp])#distance euclidienne entre observations
res2.hclust <- hclust(D,method = "ward.D2")#CAH par méthode de Ward
barplot(sort(res2.hclust$height,decreasing = TRUE)[1:15],
        names.arg = 1:15,
        xlab = "index",
        ylab = "hauteur de fusion")

#par defaut (avec 3 clusters) avec consolidation
res2.HCPC<-HCPC(res2.MCA,nb.clust=3,consol=TRUE,graph=FALSE)
plot.HCPC(res2.HCPC,choice='tree',title='Arbre hiérarchique')
#choix de 6 classes avec consolidation
res2.HCPC<-HCPC(res.MCA,nb.clust=6,consol=TRUE,graph=FALSE)
plot.HCPC(res2.HCPC,choice='tree',title='Arbre hiérarchique')
#representation axe 1:2
plot.HCPC(res2.HCPC,choice='map',draw.tree=FALSE,title='Plan factoriel')
plot.HCPC(res2.HCPC,choice='3D.map',ind.names=FALSE,centers.plot=FALSE,angle=60,
title='Arbre hiérarchique sur le plan factoriel')
#representation axes 3:4
plot.HCPC(res2.HCPC,choice='map',draw.tree=FALSE,title='Plan
factoriel',axes=c(3,4))
plot.HCPC(res2.HCPC,choice='3D.map',ind.names=FALSE,centers.plot=FALSE,angle=60,
```

```
         title='Arbre hiérarchique sur le plan factoriel',axes=c(3,4))


#resume des classes avec factoshiny
summary(res2.HCPC)
```