PCA

2023-03-16

# Set the working directory

## Read Data

#```{r, echo=FALSE} # Go to the MD project folder before running the script #dir <- “C:/Users/polcg/repos/MD” # for some reason the it can’t access current directory # Import preprocessed data # db <- read.csv(paste0(dir,“/data/Preprocessed\_data.csv”),header=T, sep=“,”, strip.white = TRUE) # take out trailing whitespaces

# read dataframe back in from RDS file

#db <- readRDS(paste0(dir,“/data/Preprocessed\_data\_rds.rds”)) #```

# Try this to read the data

## We check for missing data

## [1] 0

## [1] "age"   
## [2] "class.of.worker"   
## [3] "detailed.industry.recode"   
## [4] "detailed.occupation.recode"   
## [5] "education"   
## [6] "wage.per.hour"   
## [7] "marital.stat"   
## [8] "major.industry.code"   
## [9] "major.occupation.code"   
## [10] "race"   
## [11] "hispanic.origin"   
## [12] "sex"   
## [13] "full.or.part.time.employment.stat"   
## [14] "capital.gains"   
## [15] "capital.losses"   
## [16] "dividends.from.stocks"   
## [17] "tax.filer.stat"   
## [18] "detailed.household.and.family.stat"   
## [19] "detailed.household.summary.in.household"  
## [20] "instance.weight"   
## [21] "num.persons.worked.for.employer"   
## [22] "country.of.birth.father"   
## [23] "country.of.birth.mother"   
## [24] "country.of.birth.self"   
## [25] "citizenship"   
## [26] "veterans.benefits"   
## [27] "weeks.worked.in.year"   
## [28] "income"

## age class.of.worker   
## "integer" "factor"   
## detailed.industry.recode detailed.occupation.recode   
## "factor" "factor"   
## education wage.per.hour   
## "factor" "integer"   
## marital.stat major.industry.code   
## "factor" "factor"   
## major.occupation.code race   
## "factor" "factor"   
## hispanic.origin sex   
## "factor" "factor"   
## full.or.part.time.employment.stat capital.gains   
## "factor" "integer"   
## capital.losses dividends.from.stocks   
## "integer" "integer"   
## tax.filer.stat detailed.household.and.family.stat   
## "factor" "factor"   
## detailed.household.summary.in.household instance.weight   
## "factor" "numeric"   
## num.persons.worked.for.employer country.of.birth.father   
## "integer" "factor"   
## country.of.birth.mother country.of.birth.self   
## "factor" "factor"   
## citizenship veterans.benefits   
## "factor" "factor"   
## weeks.worked.in.year income   
## "integer" "factor"

## We separate numerical and categorical features

## age wage.per.hour   
## 1 6   
## capital.gains capital.losses   
## 14 15   
## dividends.from.stocks instance.weight   
## 16 20   
## num.persons.worked.for.employer weeks.worked.in.year   
## 21 27

## class.of.worker detailed.industry.recode   
## 2 3   
## detailed.occupation.recode education   
## 4 5   
## marital.stat major.industry.code   
## 7 8   
## major.occupation.code race   
## 9 10   
## hispanic.origin sex   
## 11 12   
## full.or.part.time.employment.stat tax.filer.stat   
## 13 17   
## detailed.household.and.family.stat detailed.household.summary.in.household   
## 18 19   
## country.of.birth.father country.of.birth.mother   
## 22 23   
## country.of.birth.self citizenship   
## 24 25   
## veterans.benefits income   
## 26 28

## Perform PCA

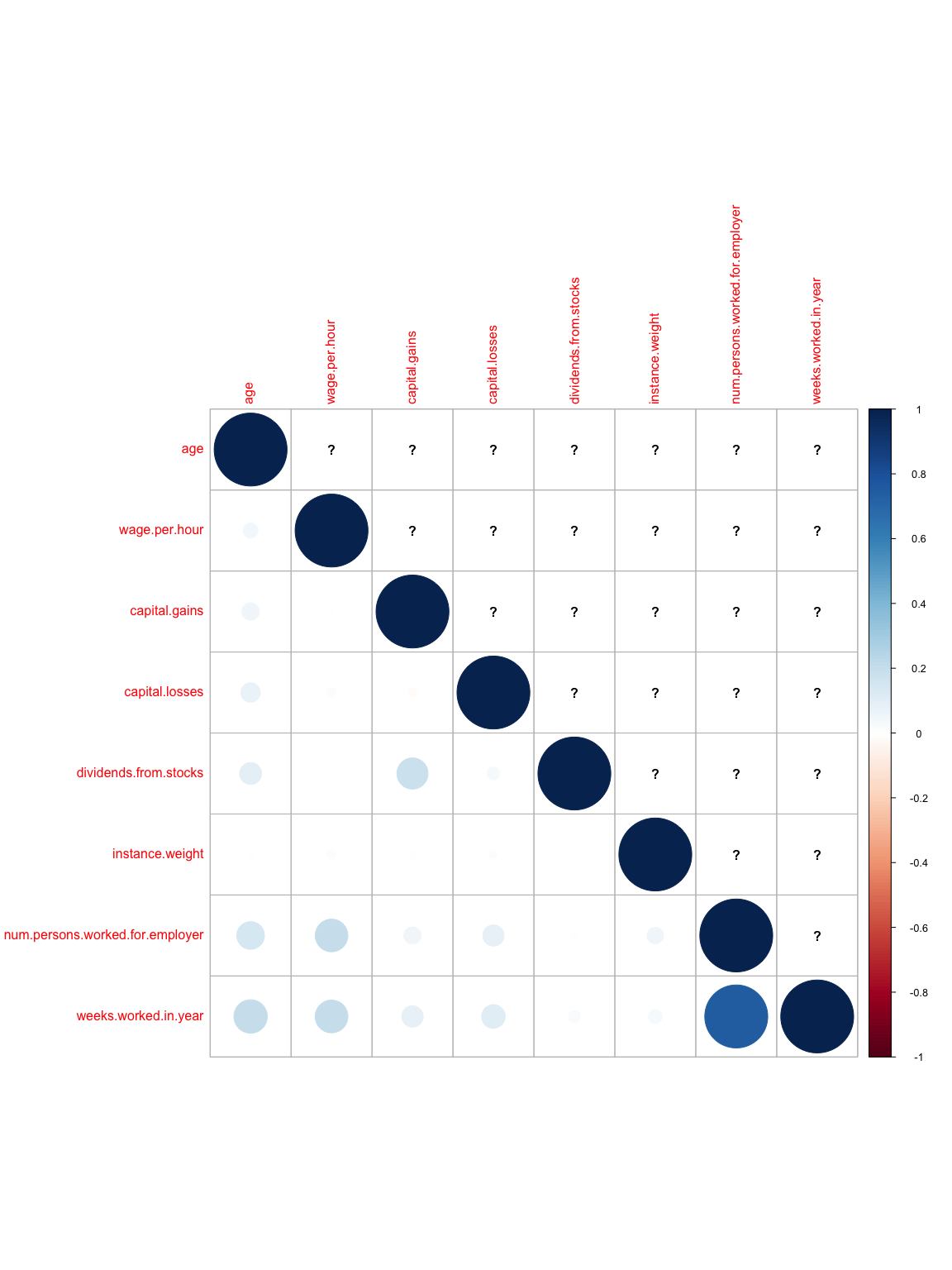
## Importance of components:  
## PC1 PC2 PC3 PC4 PC5 PC6 PC7  
## Standard deviation 1.3985 1.0942 1.0070 0.9990 0.9531 0.9393 0.89216  
## Proportion of Variance 0.2445 0.1497 0.1268 0.1247 0.1135 0.1103 0.09949  
## Cumulative Proportion 0.2445 0.3941 0.5209 0.6456 0.7592 0.8695 0.96895  
## PC8  
## Standard deviation 0.49838  
## Proportion of Variance 0.03105  
## Cumulative Proportion 1.00000

## Correlation matrix

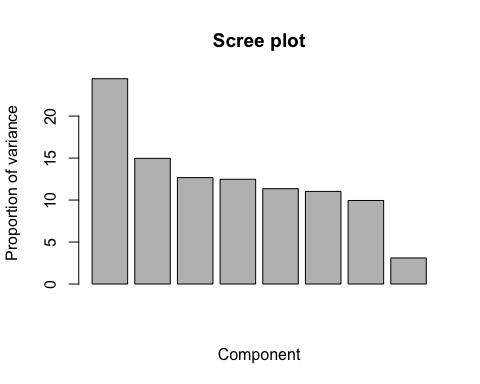
## corrplot 0.92 loaded

## Plot the correlation matrix

corrplot(cor\_mat, method="circle")

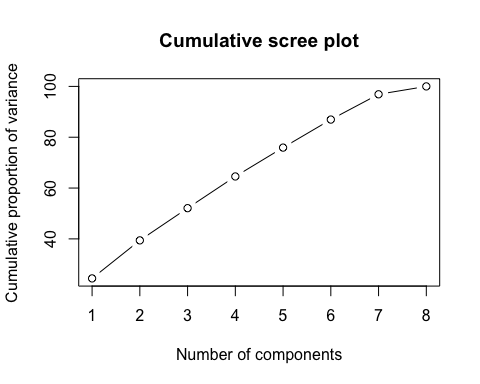


# pca$sdev  
variance<- pca$sdev^2   
# variance  
totalVariance<- sum(variance)  
# totalVariance  
percentageVar<- 100\*variance/totalVariance  
# percentageVar  
barplot(percentageVar, main="Scree plot", ylab="Proportion of variance", xlab = "Component")



## We use cumulative proportion of variance to easily spot how many components of pca to choose

cum\_prop\_var <- cumsum(percentageVar)  
plot(cum\_prop\_var, type="b", xlab="Number of components", ylab="Cumulative proportion of variance", main="Cumulative scree plot")



In this case we would choose 5

# SELECTION OF THE SINGIFICNT DIMENSIONS (keep 80% of total inertia)  
  
nd = 5  
pca$rotation # Loadings of the pca (how much of that variable is represented by the principal component)

## PC1 PC2 PC3 PC4  
## age 0.27165455 -0.26282697 -0.37222236 0.02847641  
## wage.per.hour 0.28115151 0.19136156 0.26687867 0.17411384  
## capital.gains 0.12359532 -0.62360052 0.30312730 0.05326405  
## capital.losses 0.15134175 -0.02876875 -0.73256422 -0.40133193  
## dividends.from.stocks 0.07584342 -0.68800368 0.02141987 -0.04181086  
## instance.weight 0.06233485 0.04821283 0.39156372 -0.89397569  
## num.persons.worked.for.employer 0.62468737 0.14303526 0.08176079 0.03879062  
## weeks.worked.in.year 0.63961342 0.09200473 0.03498587 0.05023641  
## PC5 PC6 PC7 PC8  
## age 0.51611330 -0.6143016 -0.26472613 -0.062817628  
## wage.per.hour -0.64575328 -0.5809296 -0.16825794 -0.002747413  
## capital.gains -0.11330970 0.3160451 -0.62267807 -0.025919591  
## capital.losses -0.47197125 0.1767846 -0.15581006 -0.016246414  
## dividends.from.stocks -0.21046712 -0.1531580 0.67151882 -0.000679995  
## instance.weight 0.11645127 -0.1637966 -0.02664231 0.013235285  
## num.persons.worked.for.employer 0.09547653 0.2489304 0.16179480 -0.695598603  
## weeks.worked.in.year 0.11857799 0.2043626 0.10853429 0.714896898

At a quick glance it would seem that the most important variables in terms of the variance be num.persons.worked.for.employer and weeks.worked.in.year since they have the higher absolute value in PC1 which amounts for the most amount of variance.

## We project our data on only 5 dimensions of pca

# STORAGE OF THE EIGENVALUES, EIGENVECTORS AND PROJECTIONS IN THE nd DIMENSIONS  
# View(pca$x)  
dim(pca$x)

## [1] 20000 8

dim(numerical\_data)

## [1] 20000 8

numerical\_data[2000,]

## age wage.per.hour capital.gains capital.losses dividends.from.stocks  
## 2000 58 0 0 0 0  
## instance.weight num.persons.worked.for.employer weeks.worked.in.year  
## 2000 1706.64 6 40

pca$x[2000,]

## PC1 PC2 PC3 PC4 PC5 PC6   
## 1.680889370 0.109800547 -0.237097498 0.186678084 1.020025087 -0.004955472   
## PC7 PC8   
## 0.121030712 -0.762899412

Psi = pca$x[,1:nd]  
dim(Psi)

## [1] 20000 5

Psi[2000,]

## PC1 PC2 PC3 PC4 PC5   
## 1.6808894 0.1098005 -0.2370975 0.1866781 1.0200251

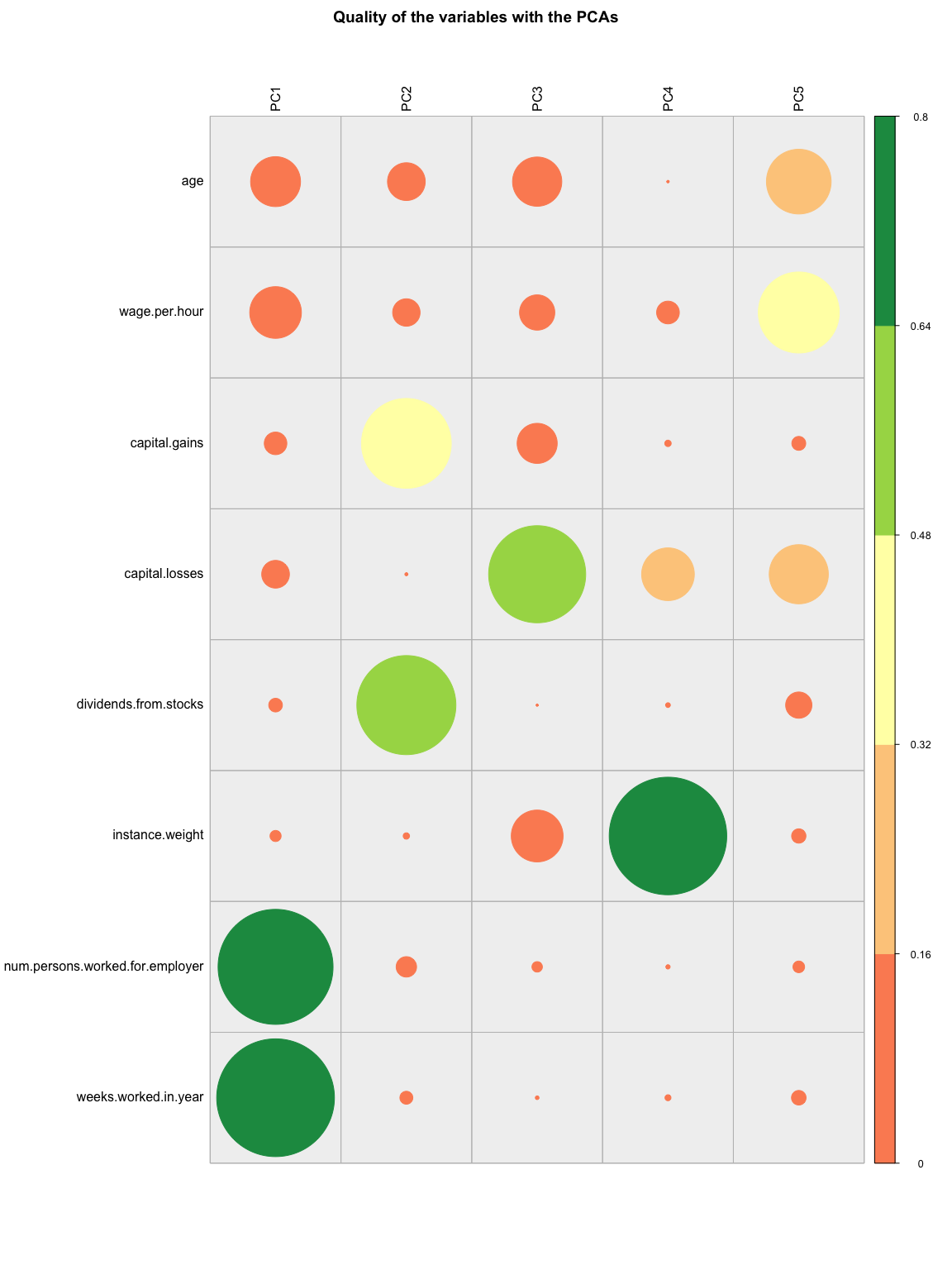
# Observations on significant PCAs.  
SignificantCoord = pca$x[,1:nd]  
CorrelationSignificant = cor(numerical\_data,SignificantCoord)

## We store some data to plot later

iden = row.names(numerical\_data)  
etiq = names(numerical\_data)  
ze = rep(0,length(etiq)) # WE WILL NEED THIS VECTOR AFTERWARDS FOR THE GRAPHICS

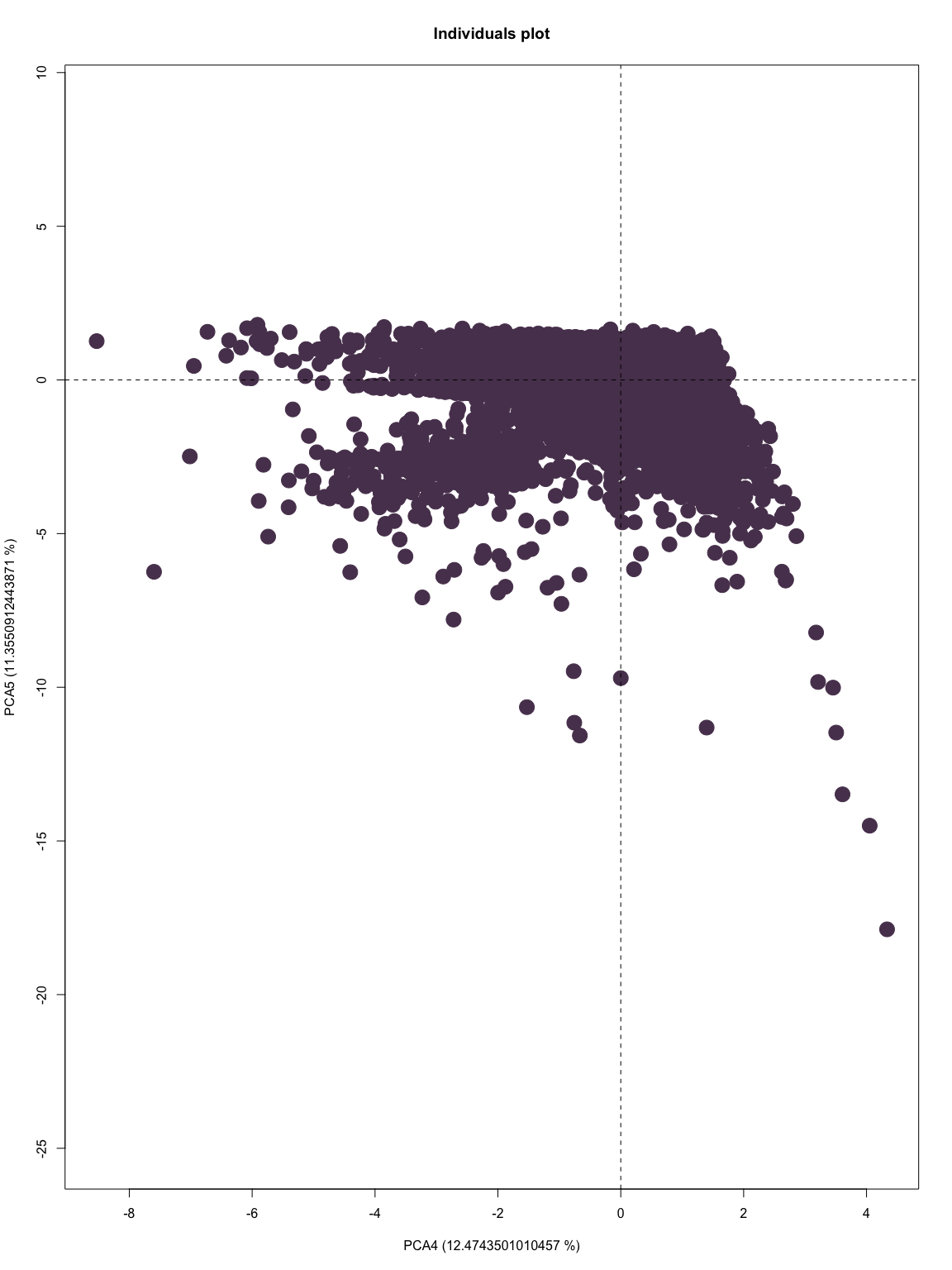
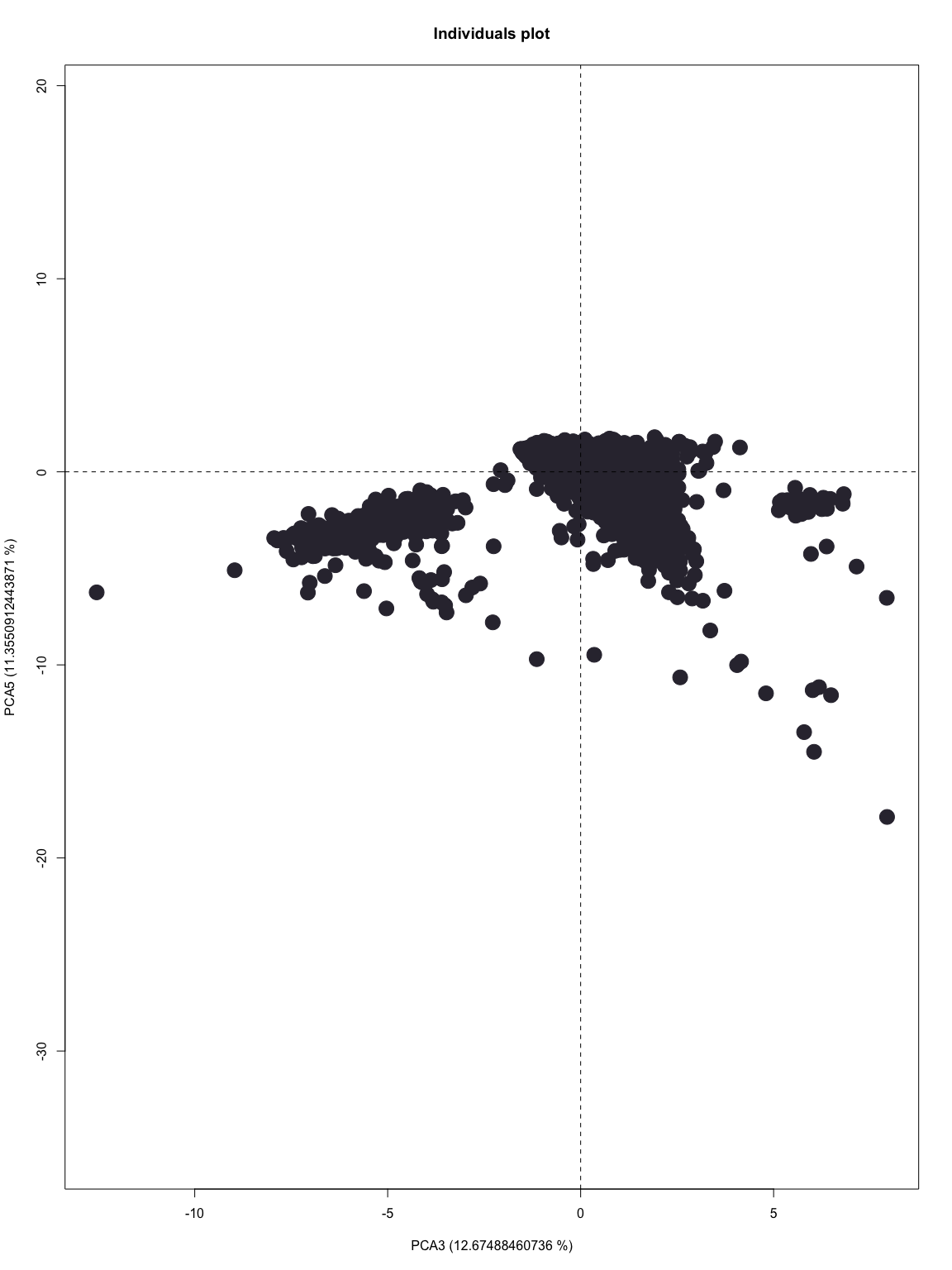
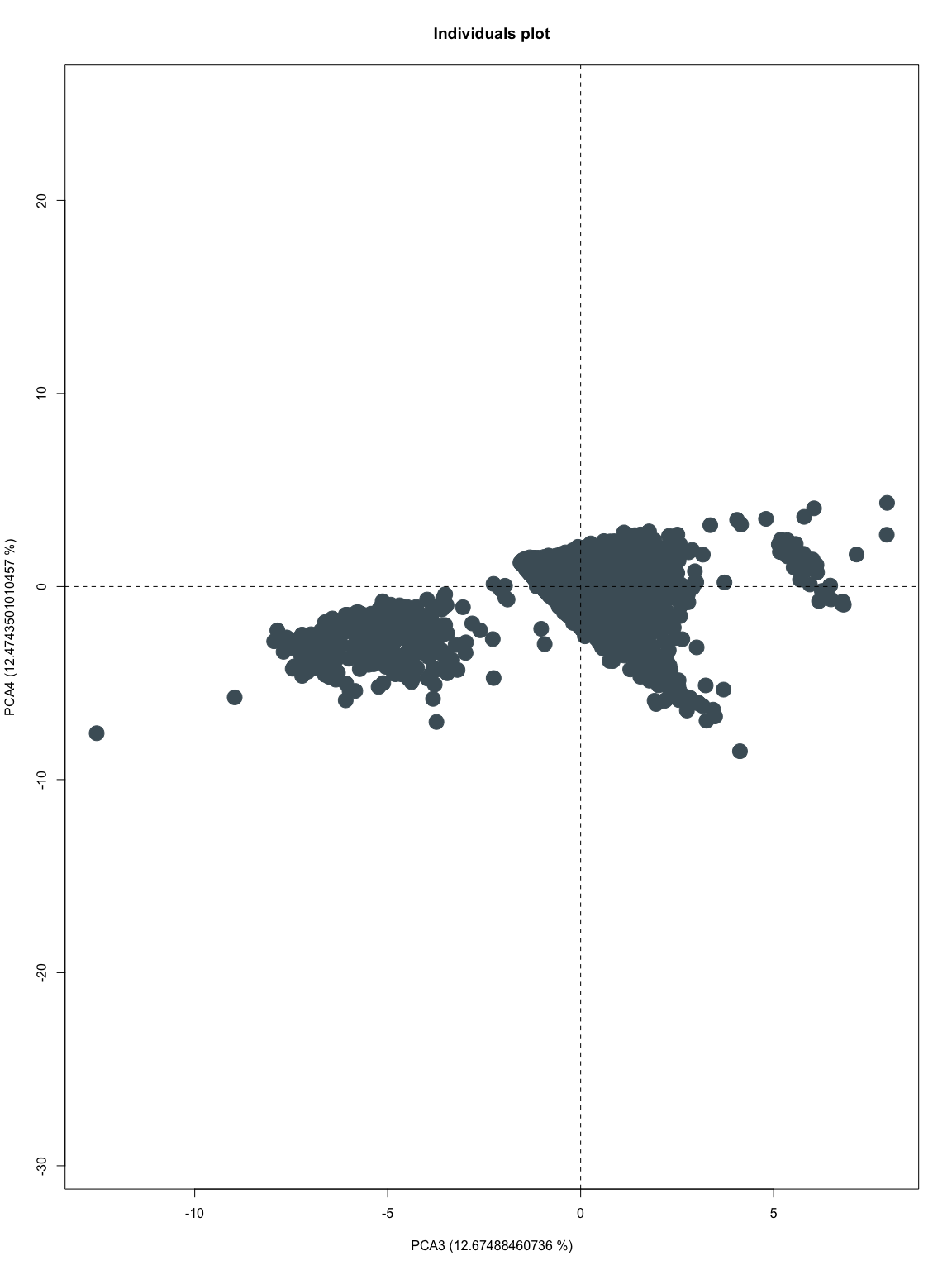
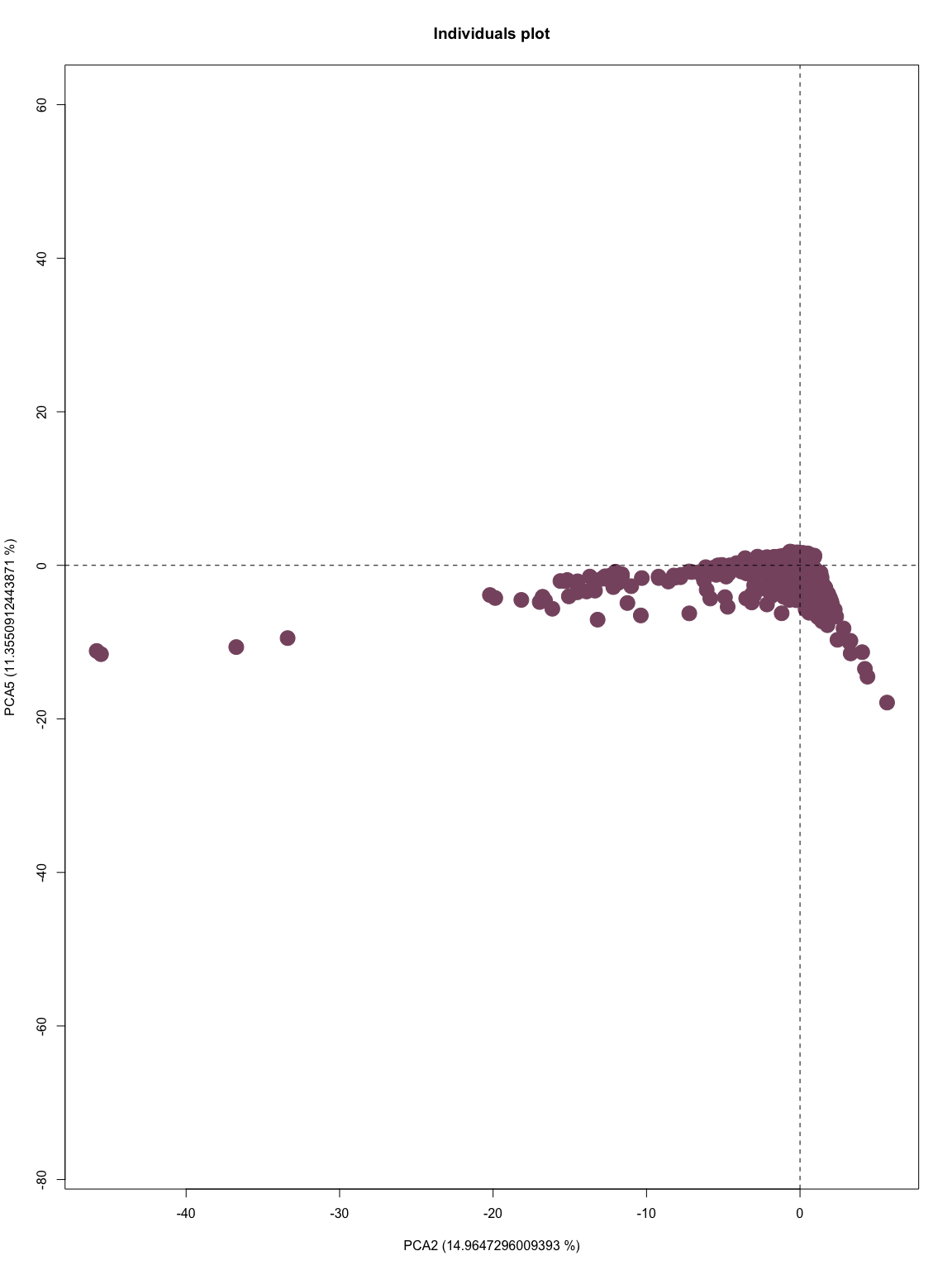
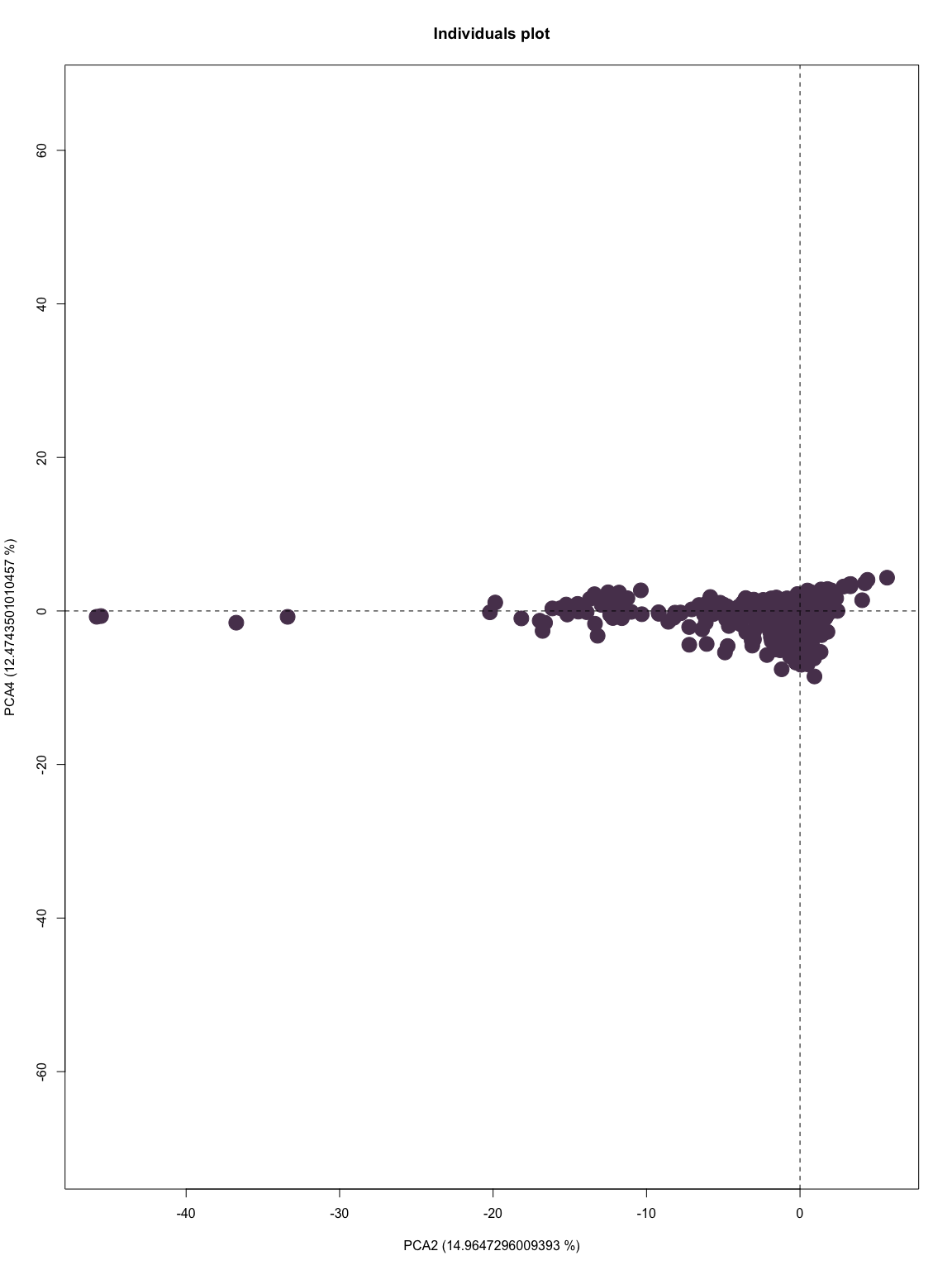
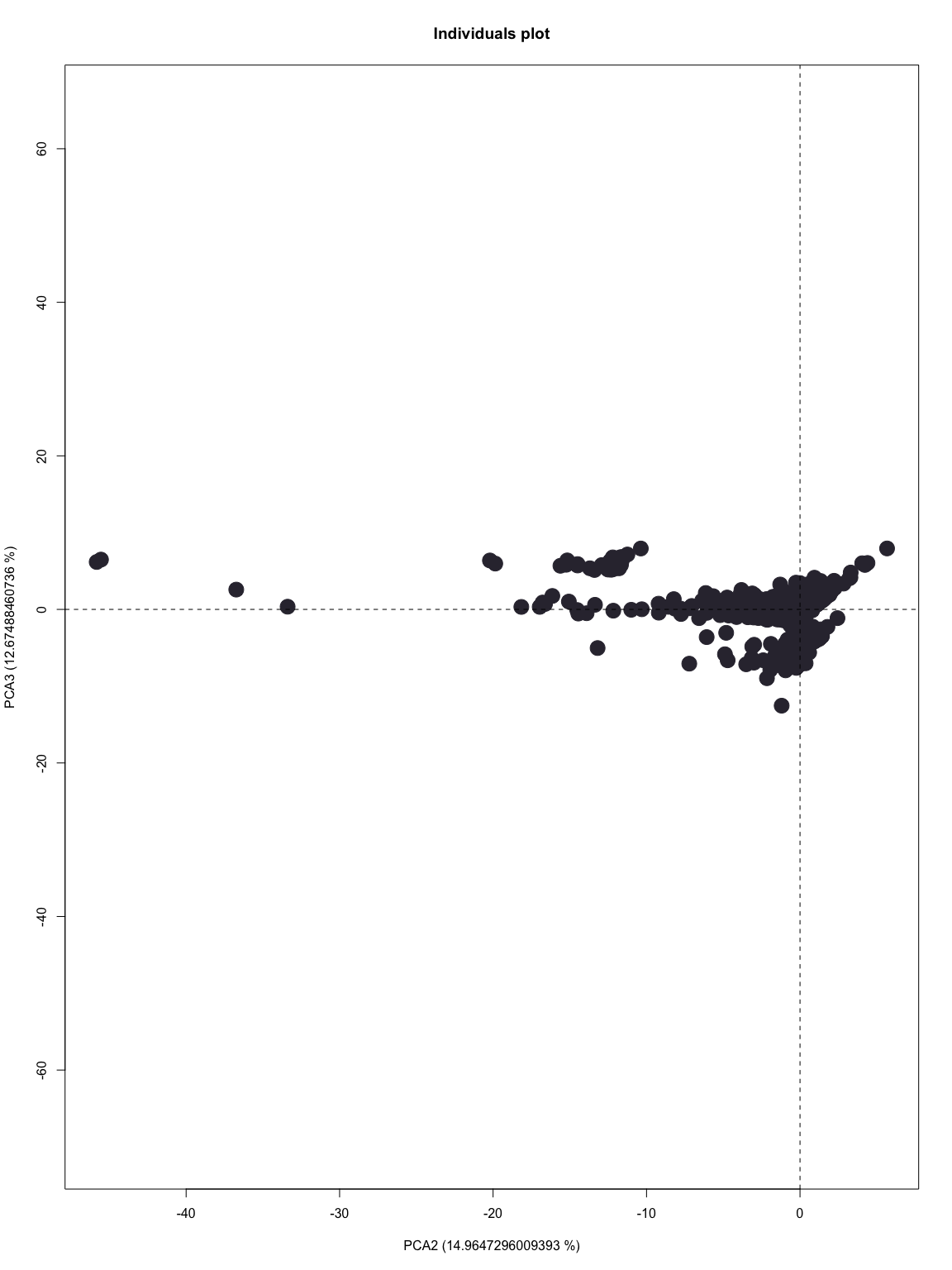
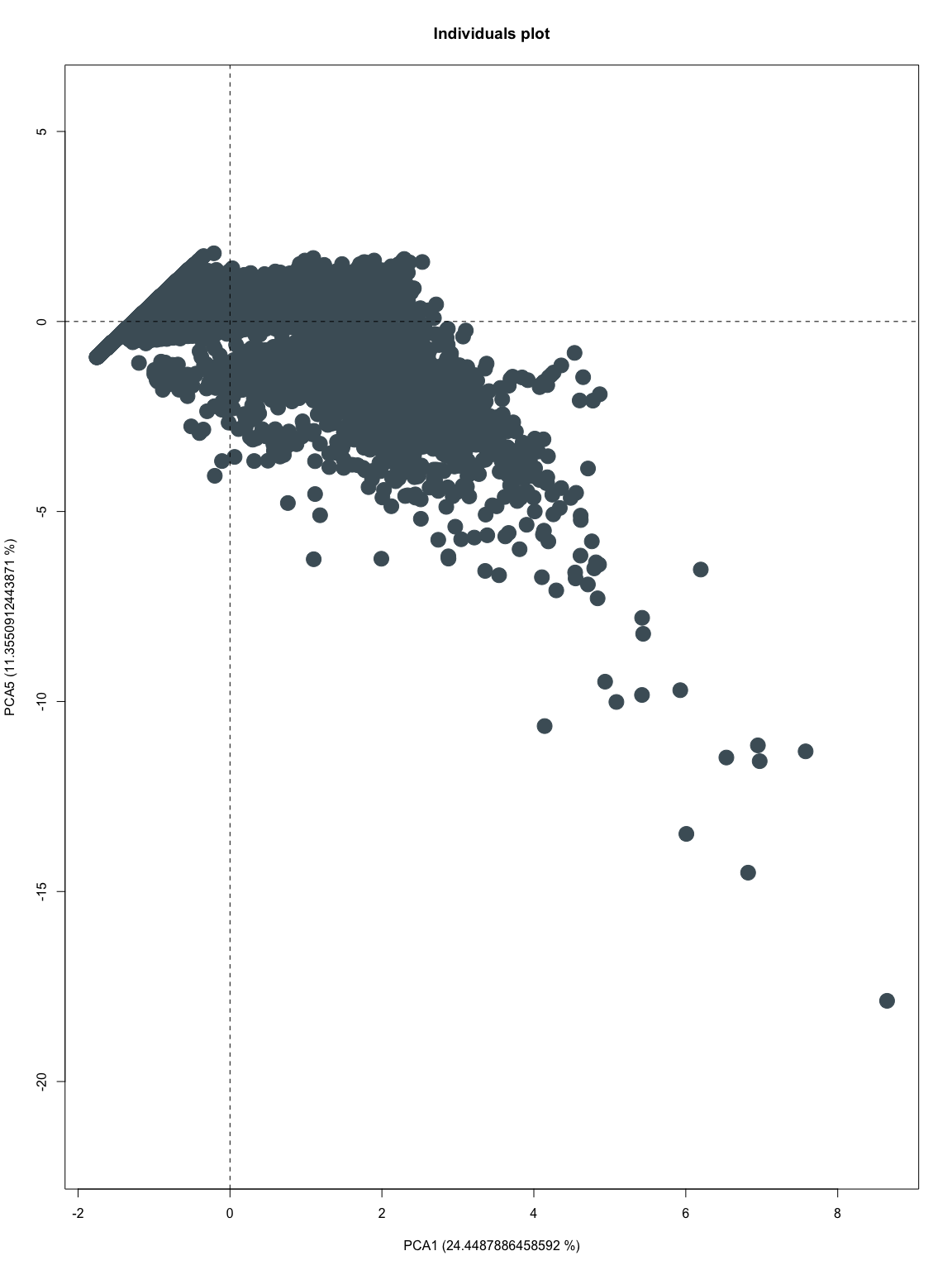
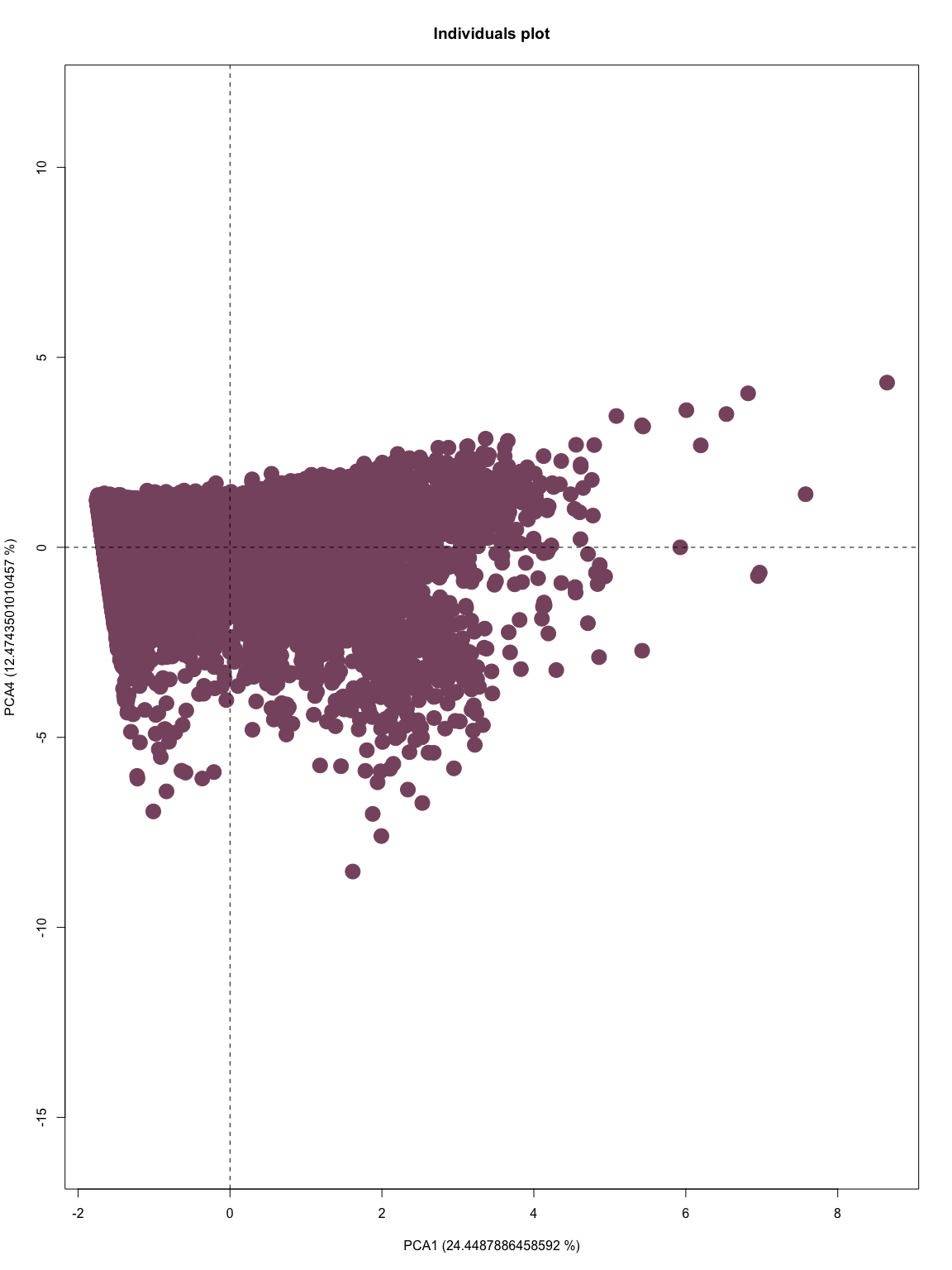
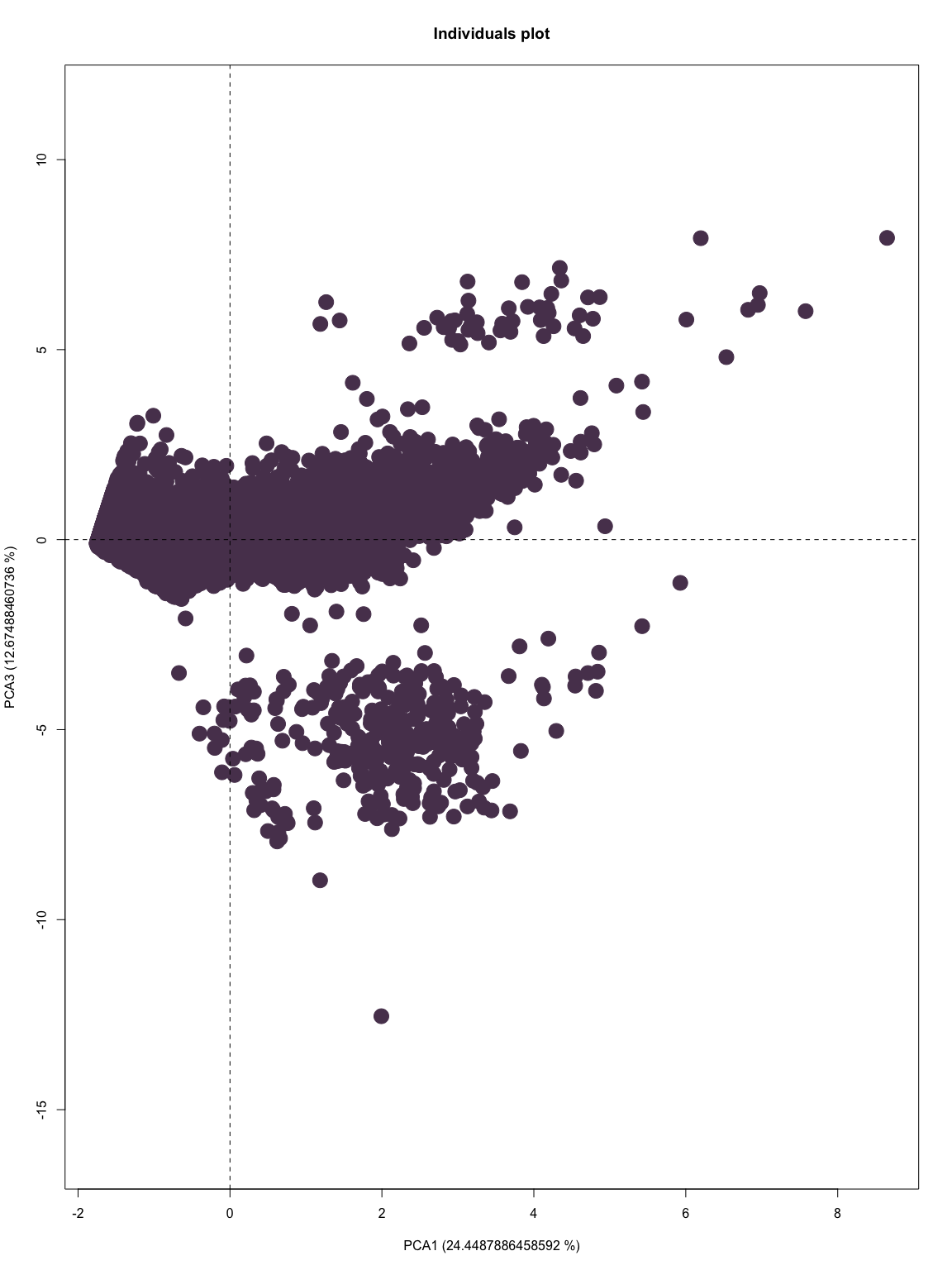
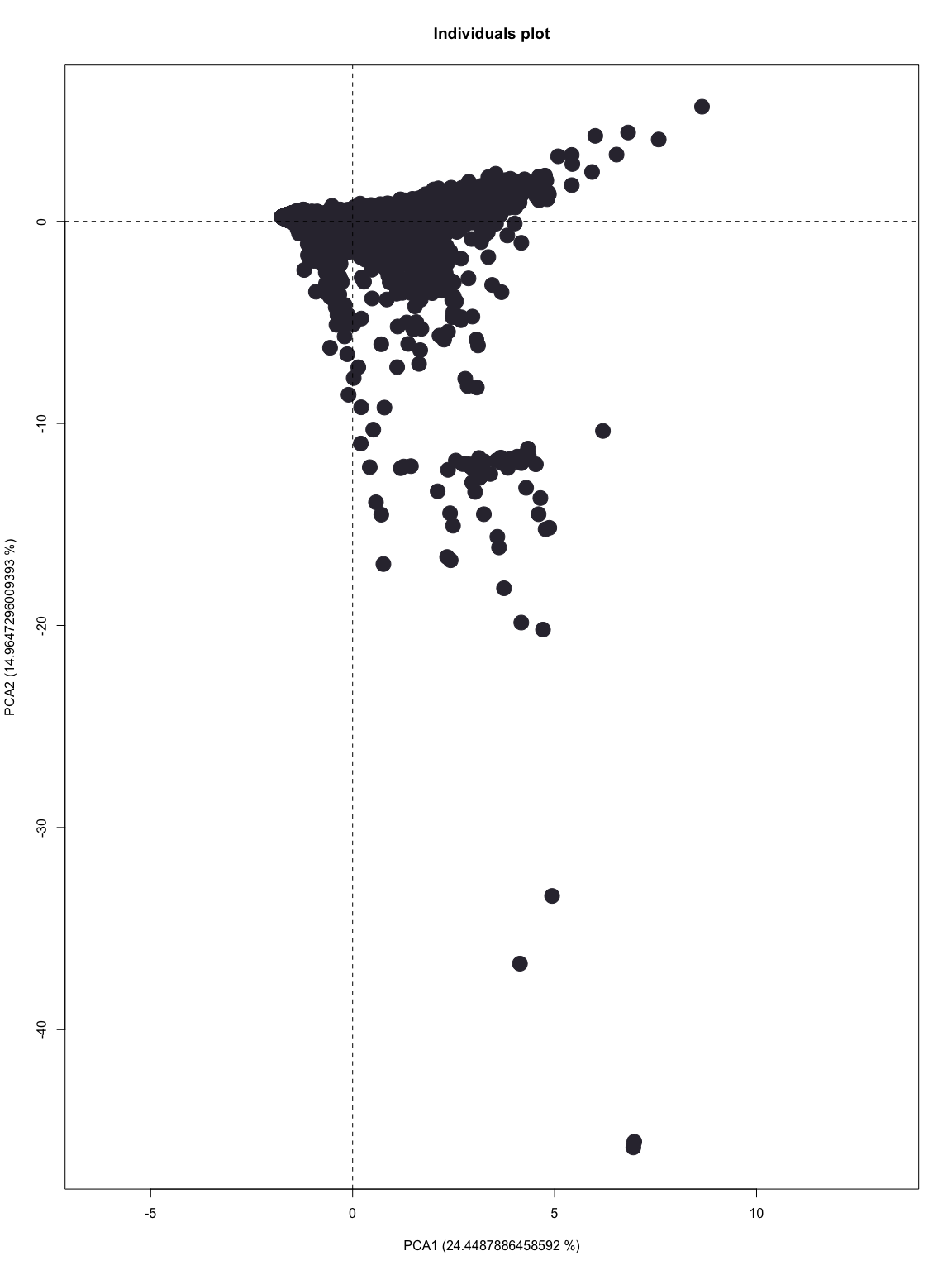
# Plot of quality of variables using the square cosine

SquareCos = CorrelationSignificant^2  
color\_palette <- c("#FC8D62", "#FDCC8A", "#FFFFB3", "#A6D854", "#1A9850")  
corrplot(SquareCos, is.corr=FALSE, title="Quality of the variables with the PCAs",col =color\_palette, bg = "#F1F1F1", tl.col = 'black', mar=c(3,0,2,0))



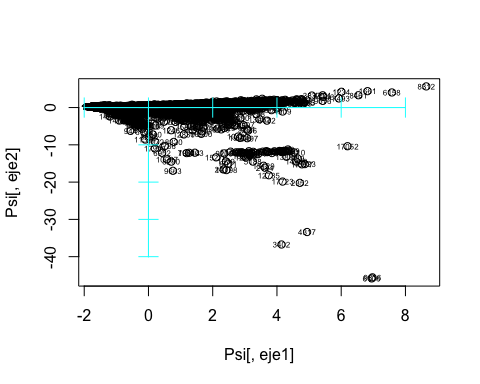
# Individuals Plot

color\_palette1 <- c("#4B5D67", "#322F3D", "#59405C", "#87556F")  
colorIteration <- 1  
  
for (i in seq(nd-1)) {  
 for (j in seq(i+1, nd)) {  
 plot(SignificantCoord[, i], SignificantCoord[, j],  
 col = color\_palette1[(colorIteration %% length(color\_palette1)) + 1],  
 main = "Individuals plot",  
 xlab = paste0("PCA", i, " (", percentageVar[i], ' %)'),  
 ylab = paste0("PCA", j, " (", percentageVar[j], ' %)'),  
 pch = 19,  
 cex = 2.5, asp = 0.5)  
 abline(h = 0, v = 0, lty = "dashed")  
   
   
 colorIteration <- colorIteration + 1  
 }  
}



# Correlation Circle

eje1<-1  
#eje2<-3  
eje2<-2  
  
plot(Psi[,eje1],Psi[,eje2])  
text(Psi[,eje1],Psi[,eje2],labels=iden, cex=0.5)  
axis(side=1, pos= 0, labels = F, col="cyan")  
axis(side=3, pos= 0, labels = F, col="cyan")  
axis(side=2, pos= 0, labels = F, col="cyan")  
axis(side=4, pos= 0, labels = F, col="cyan")



For example let’s take a look at 3402 to make sure it makes sense: as we can see PC1 corresponds roughly to 4 and PC2 is in the middle of -40 and -30

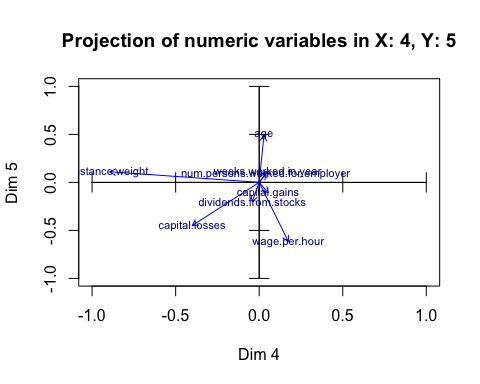
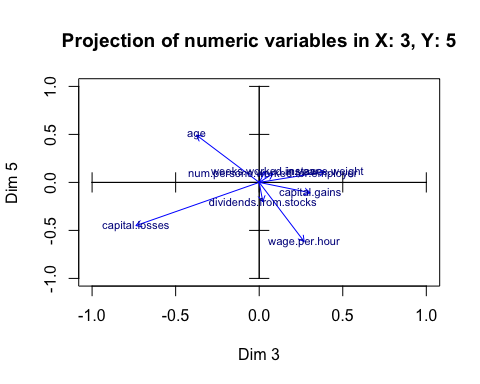
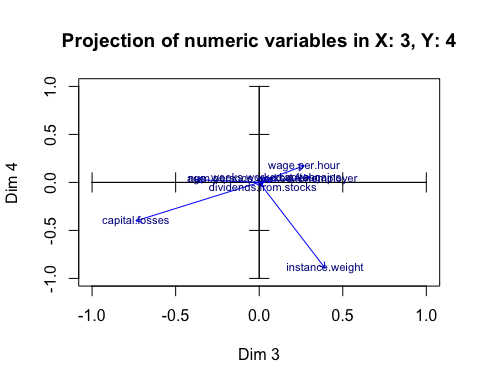
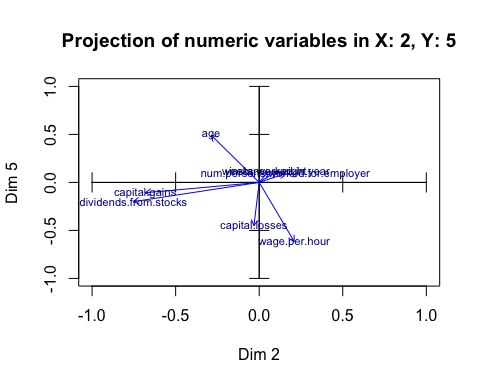
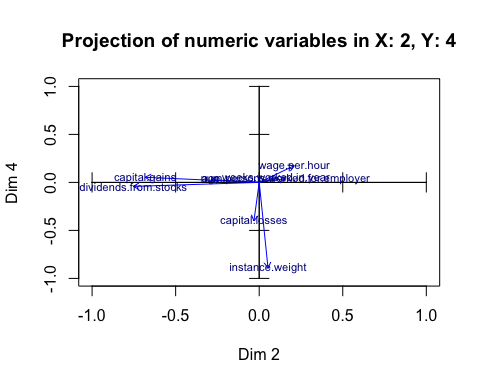
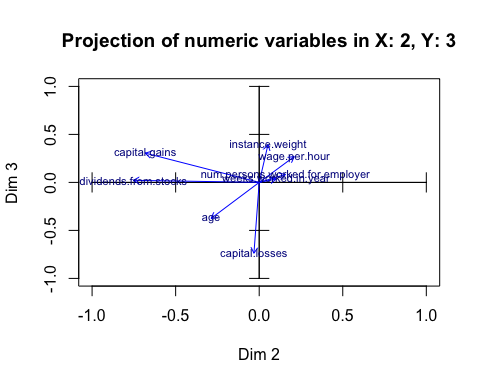
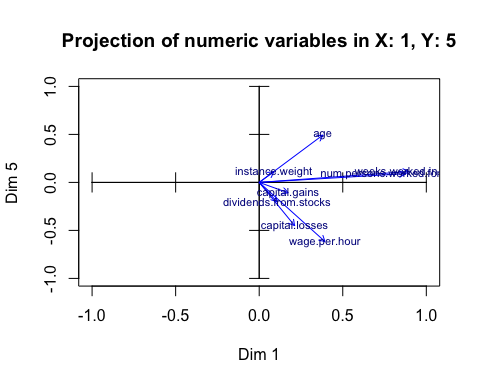
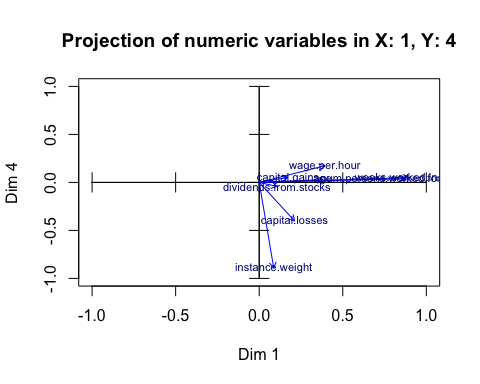
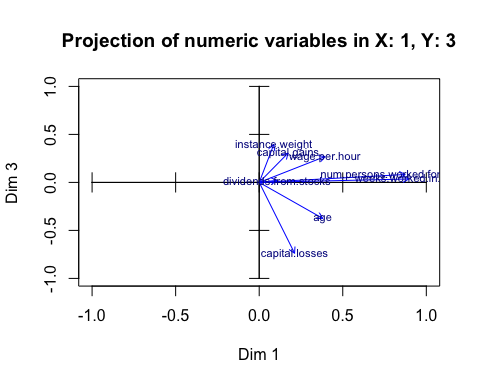
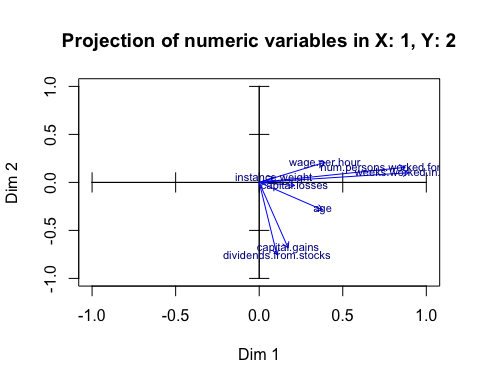
Psi[3402,]

## PC1 PC2 PC3 PC4 PC5   
## 4.141940 -36.733831 2.578435 -1.528280 -10.646693

#Projection of variables  
Phi = cor(numerical\_data,Psi)  
#View(Phi)  
  
#select your axis  
  
X<-Phi[,eje1]  
Y<-Phi[,eje2]

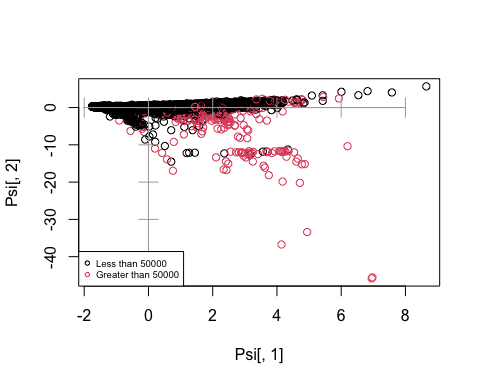
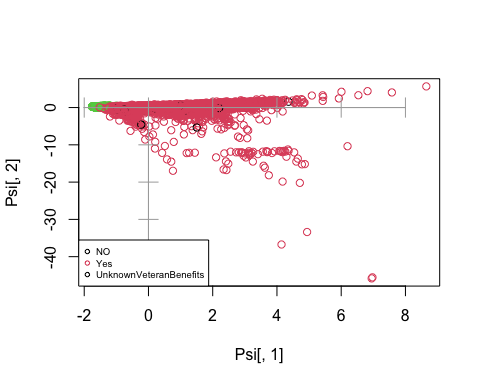
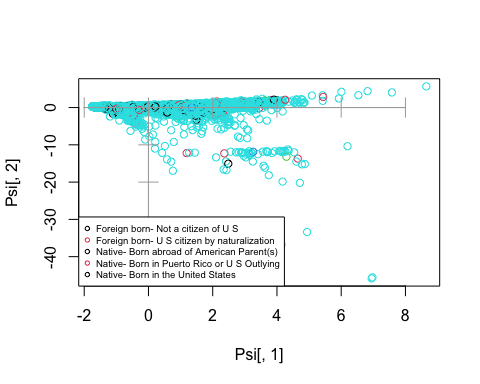
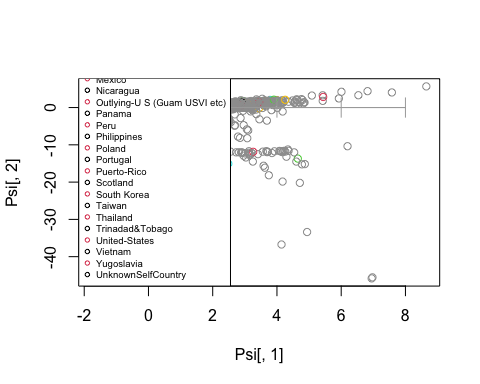
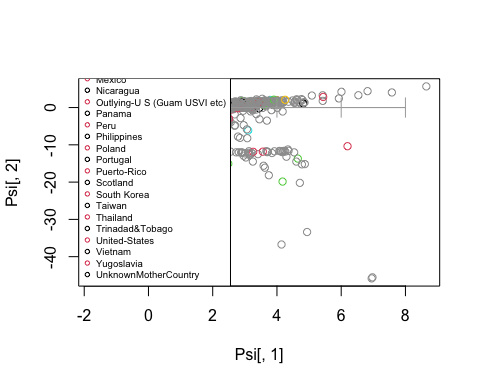
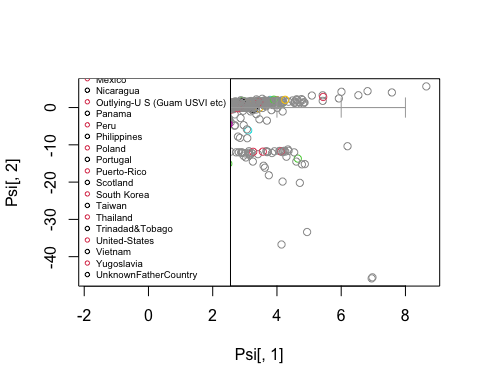
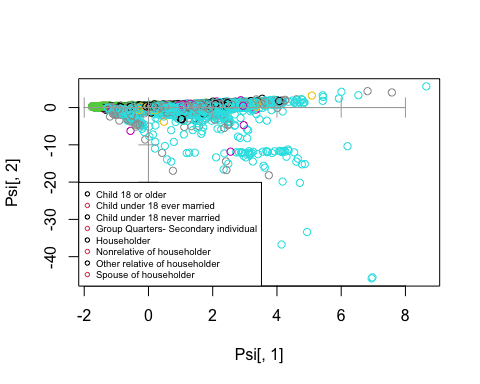
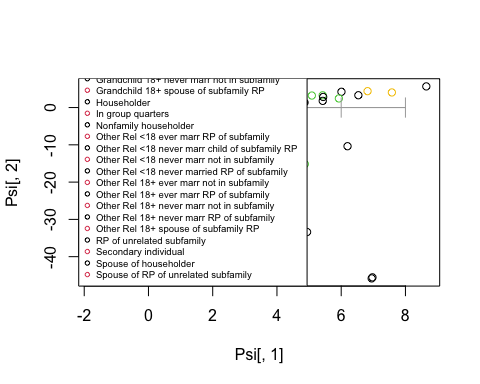
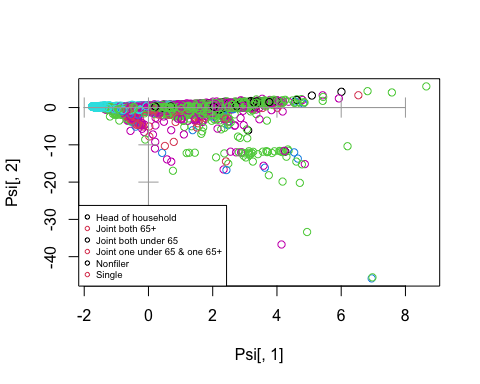
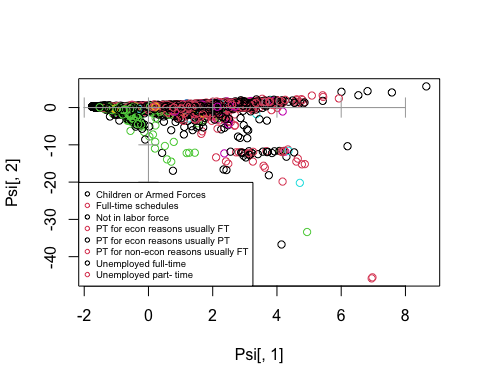
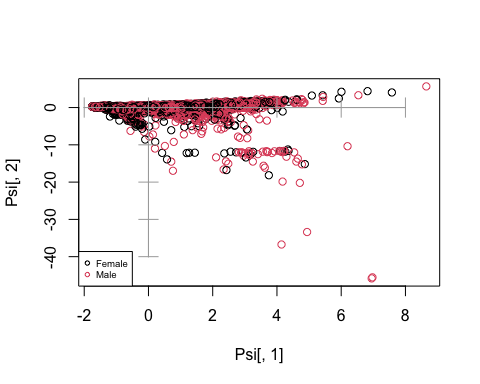
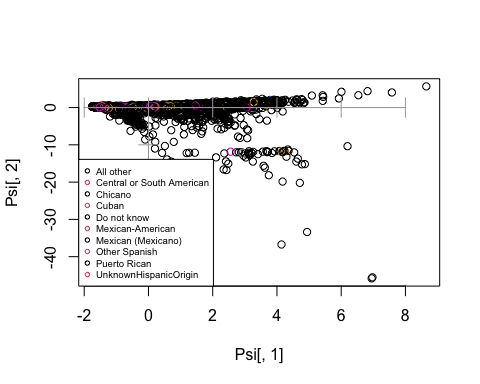
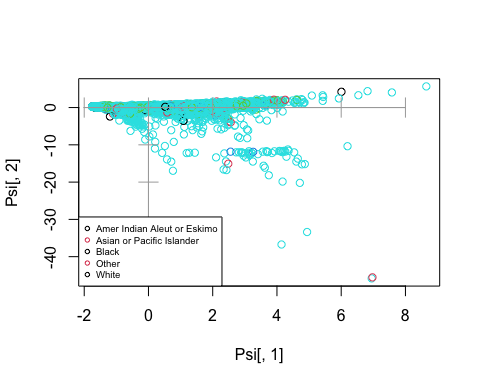
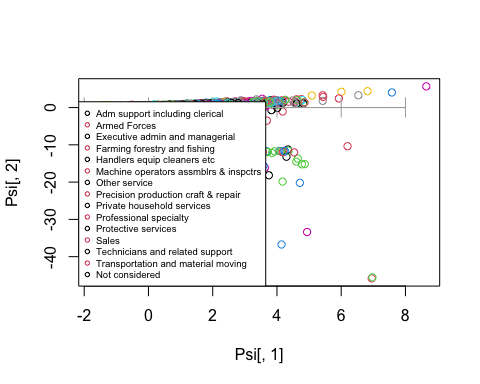
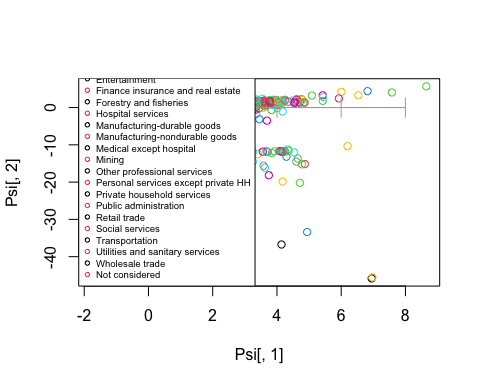
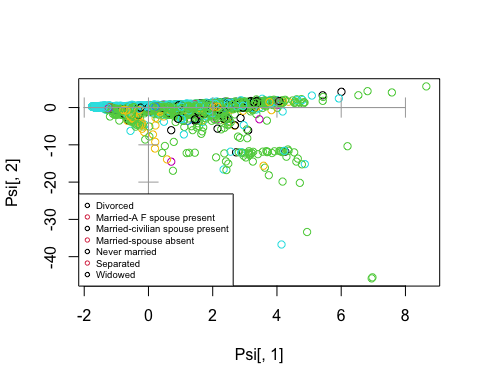
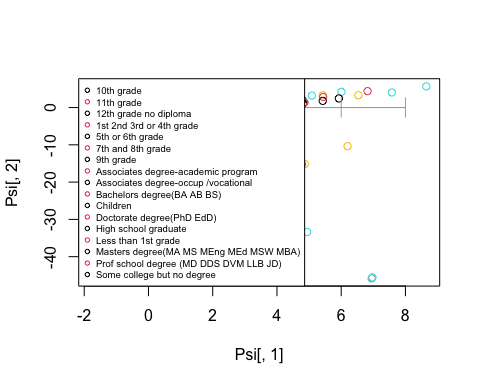
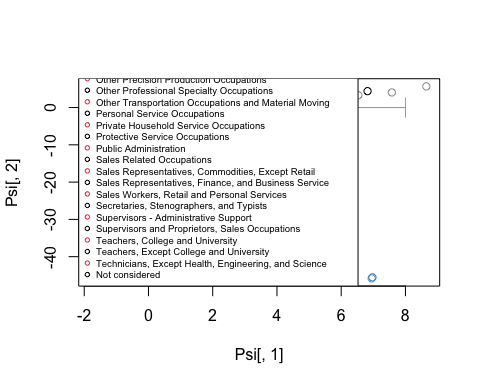
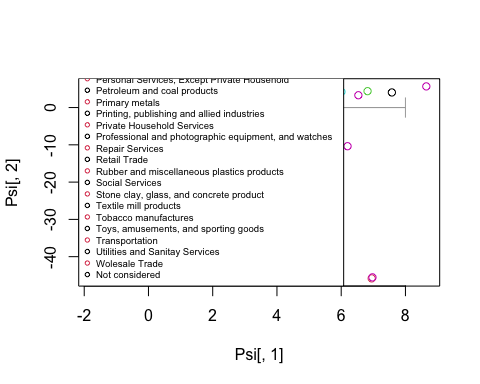
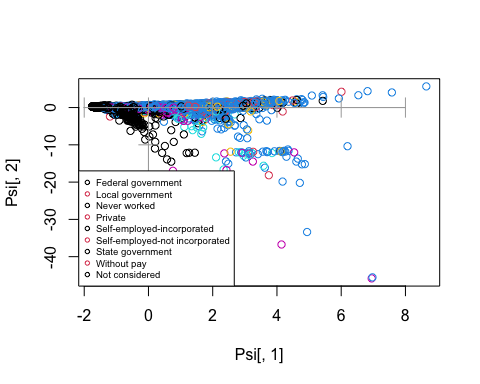
View

for (i in seq(nd-1)) {   
 for (j in seq(i+1, nd)) {  
   
 X<-Phi[,i]  
 Y<-Phi[,j]  
 plot(Psi[,i],Psi[,j],type="n", ylim = c(-1, 1), xlim = c(-1, 1), xlab = paste("Dim", i), ylab = paste("Dim", j), main = paste("Projection of numeric variables in X: ", i, ", Y: ", j, sep = "") )  
 axis(side=1, pos= 0, labels = F)  
 axis(side=3, pos= 0, labels = F)  
 axis(side=2, pos= 0, labels = F)  
 axis(side=4, pos= 0, labels = F)  
 arrows(ze, ze, X, Y, length = 0.07,col="blue")  
 text(X,Y,labels=etiq,col="darkblue", cex=0.7)  
  
 }  
}

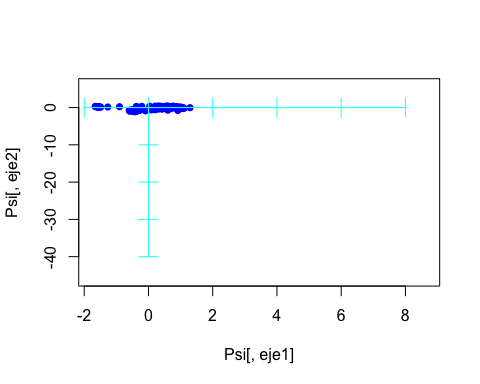


We can see something that might be interesting here, that is, there is a group of children grouped up at the same place in the following graph.

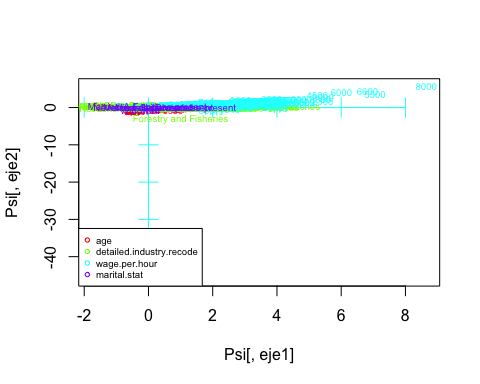
# iterate through all columns of the categorical\_data dataset  
for (i in 1:ncol(categorical\_data)) {  
 # extract the ith column  
 varcat <- factor(categorical\_data[,i])  
   
 # create a plot  
 plot(Psi[,1],Psi[,2],col=varcat)  
 axis(side=1, pos= 0, labels = F, col="darkgray")  
 axis(side=3, pos= 0, labels = F, col="darkgray")  
 axis(side=2, pos= 0, labels = F, col="darkgray")  
 axis(side=4, pos= 0, labels = F, col="darkgray")  
 legend("bottomleft",levels(factor(varcat)),pch=1,col=c(1,2), cex=0.6)  
}



#select your qualitative variable  
k<-1 #dictamen in credsco  
  
plot(Psi[,eje1],Psi[,eje2],type="n")  
varcat<-db[,k]  
fdic1 = tapply(Psi[,eje1],varcat,mean)  
fdic2 = tapply(Psi[,eje2],varcat,mean)   
  
points(fdic1,fdic2,pch=16,col="blue", labels=levels(varcat))  
text(fdic1,fdic2,labels=levels(varcat),col="blue", cex=0.7)  
  
  
#all qualitative together  
axis(side=1, pos= 0, labels = F, col="cyan")  
axis(side=3, pos= 0, labels = F, col="cyan")  
axis(side=2, pos= 0, labels = F, col="cyan")  
axis(side=4, pos= 0, labels = F, col="cyan")



#all qualitative together  
plot(Psi[,eje1],Psi[,eje2],type="n")  
axis(side=1, pos= 0, labels = F, col="cyan")  
axis(side=3, pos= 0, labels = F, col="cyan")  
axis(side=2, pos= 0, labels = F, col="cyan")  
axis(side=4, pos= 0, labels = F, col="cyan")  
#nominal qualitative variables  
  
dcat<-c(1,3,6:7)  
#divide categoricals in several graphs if joint representation saturates  
  
#build a palette with as much colors as qualitative variables   
  
#colors<-c("blue","red","green","orange","darkgreen")  
#alternative  
colors<-rainbow(length(dcat))  
  
c<-1  
for(k in dcat){  
 seguentColor<-colors[c]  
fdic1 = tapply(Psi[,eje1],db[,k],mean)  
fdic2 = tapply(Psi[,eje2],db[,k],mean)   
  
text(fdic1,fdic2,labels=levels(factor(db[,k])),col=seguentColor, cex=0.6)  
c<-c+1  
}  
legend("bottomleft",names(db)[dcat],pch=1,col=colors, cex=0.6)



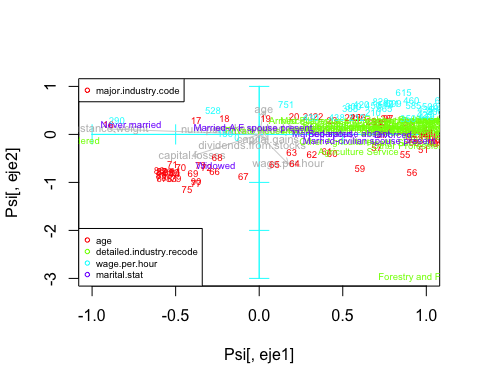
# ES UN FACTOR D'ESCALA PER DIBUIXAR LES FLETXES MES VISIBLES EN EL GRAFIC  
#fm = round(max(abs(Psi[,1])))   
fm=20  
  
#scale the projected variables  
#X<-fm\*U[,eje1]  
#Y<-fm\*U[,eje2]  
  
#represent numerical variables in background  
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(-1,1), ylim=c(-3,1))  
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))  
axis(side=1, pos= 0, labels = F, col="cyan")  
axis(side=3, pos= 0, labels = F, col="cyan")  
axis(side=2, pos= 0, labels = F, col="cyan")  
axis(side=4, pos= 0, labels = F, col="cyan")  
  
#add projections of numerical variables in background  
arrows(ze, ze, X, Y, length = 0.07,col="lightgray")  
text(X,Y,labels=etiq,col="gray", cex=0.7)  
  
#add centroids  
c<-1  
for(k in dcat){  
 seguentColor<-colors[c]  
   
 fdic1 = tapply(Psi[,eje1],db[,k],mean)  
 fdic2 = tapply(Psi[,eje2],db[,k],mean)   
   
 #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))  
 text(fdic1,fdic2,labels=levels(factor(db[,k])),col=seguentColor, cex=0.6)  
 c<-c+1  
}  
legend("bottomleft",names(db)[dcat],pch=1,col=colors, cex=0.6)  
  
  
#add ordinal qualitative variables. Ensure ordering is the correct  
  
dordi<-c(8)  
  
  
levels(factor(db[,dordi[1]]))

## [1] "Agriculture" "Armed Forces"   
## [3] "Business and repair services" "Communications"   
## [5] "Construction" "Education"   
## [7] "Entertainment" "Finance insurance and real estate"   
## [9] "Forestry and fisheries" "Hospital services"   
## [11] "Manufacturing-durable goods" "Manufacturing-nondurable goods"   
## [13] "Medical except hospital" "Mining"   
## [15] "Other professional services" "Personal services except private HH"  
## [17] "Private household services" "Public administration"   
## [19] "Retail trade" "Social services"   
## [21] "Transportation" "Utilities and sanitary services"   
## [23] "Wholesale trade" "Not considered"

#reorder modalities: when required  
db[,dordi[1]] <- factor(db[,dordi[1]], ordered=TRUE, levels= c("WorkingTypeUnknown","altres sit","temporal","fixe","autonom"))  
levels(db[,dordi[1]])

## [1] "WorkingTypeUnknown" "altres sit" "temporal"   
## [4] "fixe" "autonom"

c<-1  
col<-1  
for(k in dordi){  
 seguentColor<-colors[col]  
 fdic1 = tapply(Psi[,eje1],db[,k],mean)  
 fdic2 = tapply(Psi[,eje2],db[,k],mean)   
   
 #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))  
 #connect modalities of qualitative variables  
 lines(fdic1,fdic2,pch=16,col=seguentColor)  
 text(fdic1,fdic2,labels=levels(db[,k]),col=seguentColor, cex=0.6)  
 c<-c+1  
 col<-col+1  
}  
legend("topleft",names(db)[dordi],pch=1,col=colors[1:length(dordi)], cex=0.6)



#using our own colors palette  
# search palettes in internet. One might be https://r-charts.com/es/colores/  
  
colors<-c("red", "blue", "darkgreen", "orange", "violet", "magenta", "pink")  
  
#represent numerical variables in background

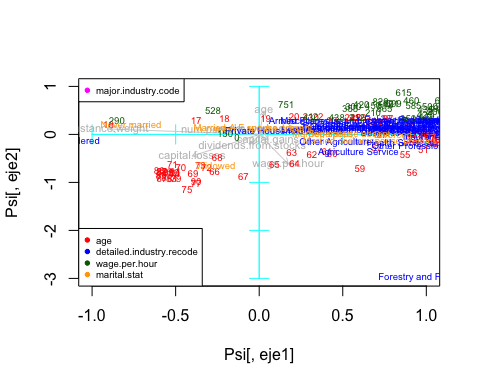
#represent numerical variables in background  
plot(Psi[,eje1],Psi[,eje2],type="n",xlim=c(-1,1), ylim=c(-3,1))  
#plot(X,Y,type="none",xlim=c(min(X,0),max(X,0)))  
axis(side=1, pos= 0, labels = F, col="cyan")  
axis(side=3, pos= 0, labels = F, col="cyan")  
axis(side=2, pos= 0, labels = F, col="cyan")  
axis(side=4, pos= 0, labels = F, col="cyan")  
  
#add projections of numerical variables in background  
arrows(ze, ze, X, Y, length = 0.07,col="lightgray")  
text(X,Y,labels=etiq,col="gray", cex=0.7)  
  
#add centroids  
c<-1  
for(k in dcat){  
 seguentColor<-colors[c]  
   
 fdic1 = tapply(Psi[,eje1],db[,k],mean)  
 fdic2 = tapply(Psi[,eje2],db[,k],mean)   
   
 #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))  
 text(fdic1,fdic2,labels=levels(factor(db[,k])),col=seguentColor, cex=0.6)  
 c<-c+1  
}  
legend("bottomleft",names(db)[dcat],pch=19,col=colors, cex=0.6)  
  
  
#add ordinal qualitative variables. Ensure ordering is the correct  
  
dordi<-c(8)  
  
  
levels(factor(db[,dordi[1]]))

## character(0)

#reorder modalities: when required  
db[,dordi[1]] <- factor(db[,dordi[1]], ordered=TRUE, levels= c("WorkingTypeUnknown","altres sit","temporal","fixe","autonom"))  
levels(db[,dordi[1]])

## [1] "WorkingTypeUnknown" "altres sit" "temporal"   
## [4] "fixe" "autonom"

c<-1  
col<-length(dcat)+1  
for(k in dordi){  
 seguentColor<-colors[col]  
 fdic1 = tapply(Psi[,eje1],db[,k],mean)  
 fdic2 = tapply(Psi[,eje2],db[,k],mean)   
   
 #points(fdic1,fdic2,pch=16,col=seguentColor, labels=levels(dd[,k]))  
 #connect modalities of qualitative variables  
 lines(fdic1,fdic2,pch=16,col=seguentColor)  
 text(fdic1,fdic2,labels=levels(db[,k]),col=seguentColor, cex=0.6)  
 c<-c+1  
 col<-col+1  
}  
legend("topleft",names(db)[dordi],pch=19,col=colors[col:col+length(dordi)-1], cex=0.6)



#Make two complementary factorial maps  
  
colors<-c("red", "blue", "darkgreen", "orange", "violet", "magenta", "pink")

This shows how much the numerical values vary from the mean

# PROJECTION OF ILLUSTRATIVE qualitative variables on individuals' map  
# PROJECCI? OF INDIVIDUALS DIFFERENTIATING THE Dictamen  
# (we need a numeric Dictamen to color)  
  
varcat=factor(db[,1])  
plot(Psi[,1],Psi[,2],col=varcat)  
axis(side=1, pos= 0, labels = F, col="darkgray")  
axis(side=3, pos= 0, labels = F, col="darkgray")  
axis(side=2, pos= 0, labels = F, col="darkgray")  
axis(side=4, pos= 0, labels = F, col="darkgray")  
legend("bottomleft",levels(varcat),pch=1,col=c(1,2), cex=0.6)  
  
  
# Overproject THE CDG OF LEVELS OF varcat  
fdic1 = tapply(Psi[,1],varcat,mean)  
fdic2 = tapply(Psi[,2],varcat,mean)   
  
text(fdic1,fdic2,labels=levels(factor(varcat)),col="cyan", cex=0.75)

