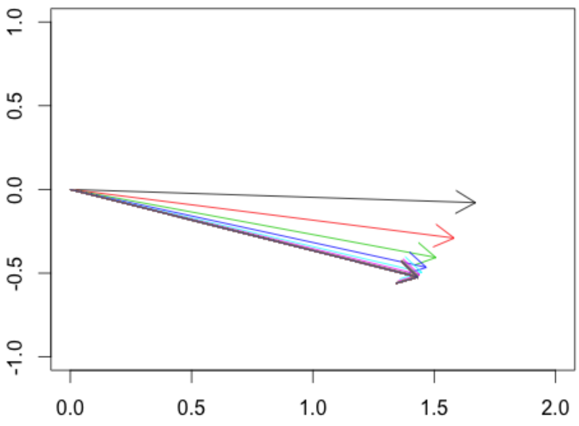
MVA deliverable 3 by Diego Garcia-Olano

1 MVA-3-Practice beyond PCA

```
1. Read Russet data set and impute the missing values of matrix formed by continuous variables.

Russet <- read.csv("/Users/diego/Documents/UPC-MIRI/semester2/MultiVariate-Analysis/
  assignment_2/Russet_ineqdata.csv", header = TRUE, quote = "\"", dec = ".", check.names=TRUE)
Russet$demo <- as.factor(Russet$demo)
levels(Russet$demo) <- c("stable","instable","dictatorship")
X <- Russet[,2:9]  #only take continous variables ( and no id)
library(mice)     #impute values using mice for "Rent" variable which has 3 NA's  and for ecks as well
X = as.matrix(complete(mice(X,m=1,seed=8675309)))
X.post.impute <- X
n = nrow(X)
p = ncol(X)
weights = rep(1,n)
N = diag(weights/sum(weights))
G = t(X) %*% N %*% rep(1,n)                     #compute centroid, colMeans()
Xc = X - rep(1,n) %*% t(G)                      #center data, scale(X,scale=FALSE)
V = t(Xc) %*% N %*% as.matrix(Xc)               #compute variance of centered matrix
                                                %V = var(Xc) * (n-1)/n  | testV = V * n/(n-1)
Xs = as.matrix(Xc)  %*% diag(sqrt(diag(V))^(-1))    #standardize data
eigX = eigen(t(Xs) %*% N %*% Xs)                 #to confirm, would get me U

2. Obtain the first three Principals Components using the NIPALS algorithm.
X = Xs
Psi = rowMeans(X)
plot(X,type="n",xlim=c(0,2),ylim=c(-1,1))
for (i in 1:100) {
  u <- t(X) %*% Psi;                       # u_h = X'_h-1 * Psi_h
  u <- u/sqrt(sum(u*u));                   # make u of unit length
  Psi <- X %*% u;                          # Psi_h = Xh-1 * u_h
  arrows(0,0,Psi[1],Psi[2],col=i)
}
sqrt(sum(Psi*Psi))    #11.87056 is first eigenvalue, and Psi is first eigenvector
```

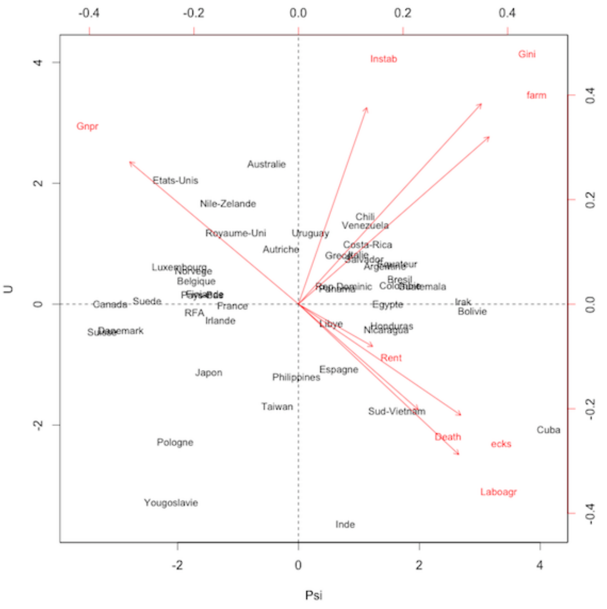


```
#Now get 2nd component
Xone = X - (Psi %*% t(u))
```

```
Psi_two = rowMeans(Xone)
for (i in 1:100) {
  u2 = t(Xone) %*% Psi_two
  u2 <- u2/sqrt(sum(u2*u2));
  Psi_two <- Xone %*% u2;
}
sqrt(sum(Psi_two*Psi_two))  #8.340813 is 2nd eigenvalue and Psi_two is second eigenvector

#Now get 2nd component by adding u as new row and Psi as new column to X
Xtwo = Xone - (Psi_two %*% t(u2))
Psi_three = rowMeans(Xtwo)
for (i in 1:100)
{
  u3 = t(Xtwo) %*% Psi_three
  u3 <- u3/sqrt(sum(u3*u3));
  Psi_three <- Xtwo %*% u3;
}
sqrt(sum(Psi_three*Psi_three))  #7.816789 is 3rd eigenvalue and Psi_three is third eigenvector

#To confirm we compute eigenvalues we would have got by diagnolizing the covariance matrix X'X
psi = eigen(t(X)%*%X)
sqrt(psi$values[1])   #11.87056 lambda1
sqrt(psi$values[2])   #8.34081  lambda2
sqrt(psi$values[3])   #7.816789 lambda3
```

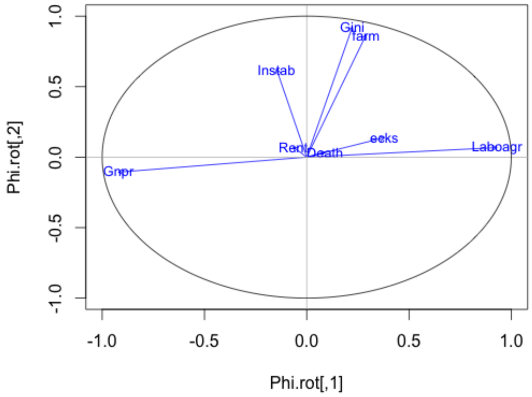
3. With the results of the NIPALS, obtain the biplot of Rp. Interpret the results.

```
U = as.matrix(cbind(u,u2,u3))                        #gather eigenvectors, U = psi$vectors[,1:3]
Psis = as.matrix(cbind(Psi,Psi_two,Psi_three))    #gather individual projections, Psis = X %*% U
rownames(Psis) = Russet$pais
rownames(U) = colnames(Russet)[2:9]
biplot(Psis,U, cex=.8, xlab="Psi",ylab="U")
abline(h=0,v=0,lty=2)
```



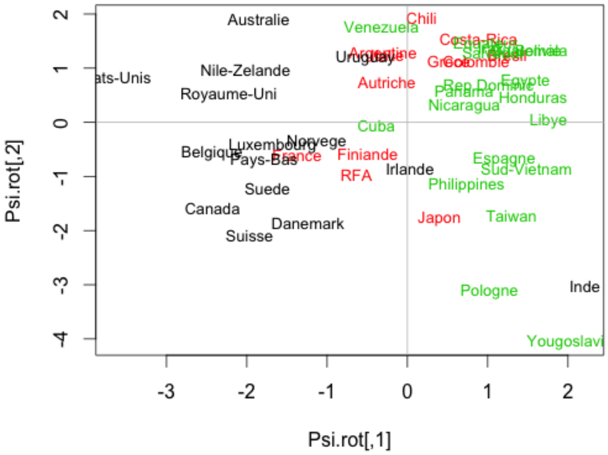
This plot is shows how Gnpr is completely negatively correlated with ecks, Death and Laboagr, and how Gini, farm, and Instab are also related with one another. Thus, countries in the northwest quadrant (Etats Unis, Australie, Royaume-Uni,etc) have higher Gnpr values and lower values for Death, Laboagr, and ecks, where as the opposite is true for countries in the southeast quadrant (Cuba, and Sud-Vietnam)

```
4. Perform the Varimax rotation, plot the rotated variables, and interpret the new rotated components.
Phi = cor(X,Psis)
pc.rot = varimax(Phi)
Phi.rot = pc.rot$loadings[1:p,]
lmb.rot = diag(t(pc.rot$loadings) %*% pc.rot$loadings)
sum(lmb.rot); sum(res.pca$eig$eigenvalue[1:3])   #both give 5.778332

library(calibrate)
ze = rep(0,p)
plot(Phi.rot,type="n",xlim=c(-1,1),ylim=c(-1,1))
text(Phi.rot,labels=colnames(Russet)[2:9], col="blue",cex=.8)
arrows(ze, ze, Phi.rot[,1], Phi.rot[,2], length = 0.07,col="blue")
abline(h=0,v=0,col="gray"); circle(1)
```



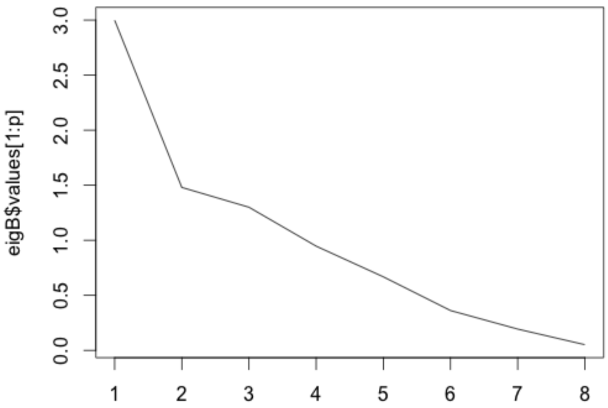
The axises have now been rotated so that Gnpr and Laboagr are along the x-axis, and so Instab, Gini and farm are more or less along the second axis. This now makes the plot a little easier to interpret.

```
5. Plot the individuals in the rotated components.
Psi_stan.rot = Xs %*% solve(cor(X.post.impute)) %*% Phi.rot
Psi.rot = Psi_stan.rot %*% diag(sqrt(lmb.rot))

plot(Psi.rot,type="n")
text(Psi.rot,labels=Russet$pais,col=as.numeric(Russet$demo),cex=.8)
abline(h=0,v=0,col="gray")
```

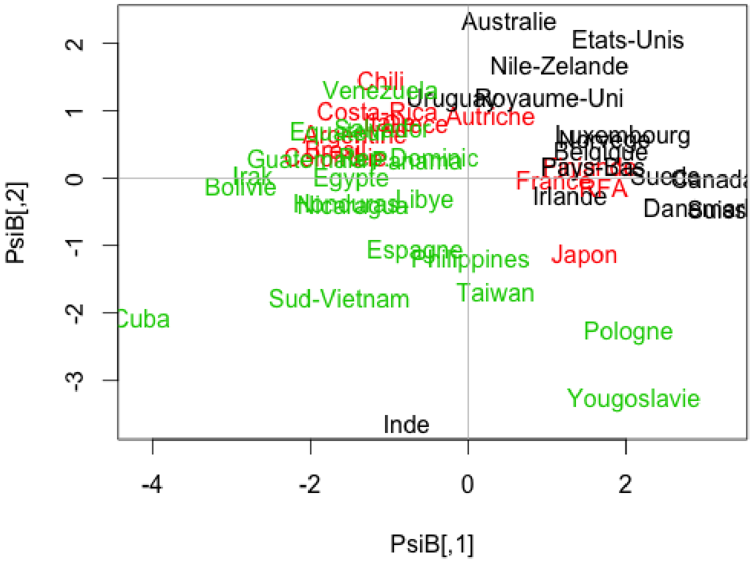


```
6. Compute the distance matrix between countries using function dist.
```

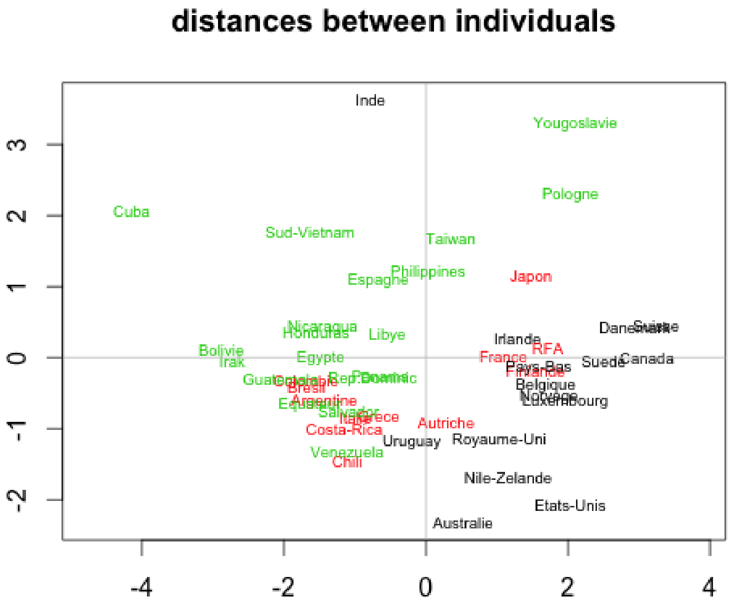
```
Obtain the mapping of countries from the matrix of distances as input.
D = dist(Xs)
D2 = (as.matrix(D))^2
Delta = diag(rep(1,n)) - (weights %*% t(weights)/sum(weights^2))
B = -(Delta %*% D2 %*% Delta)/2
eigB = eigen(N^(0.5) %*% B %*% N^(0.5))   #eigB is diagnolization of N^.5 B N^.5

plot(eigB$values[1:p],type="l")           #Show screeplot of eigenvalues
```



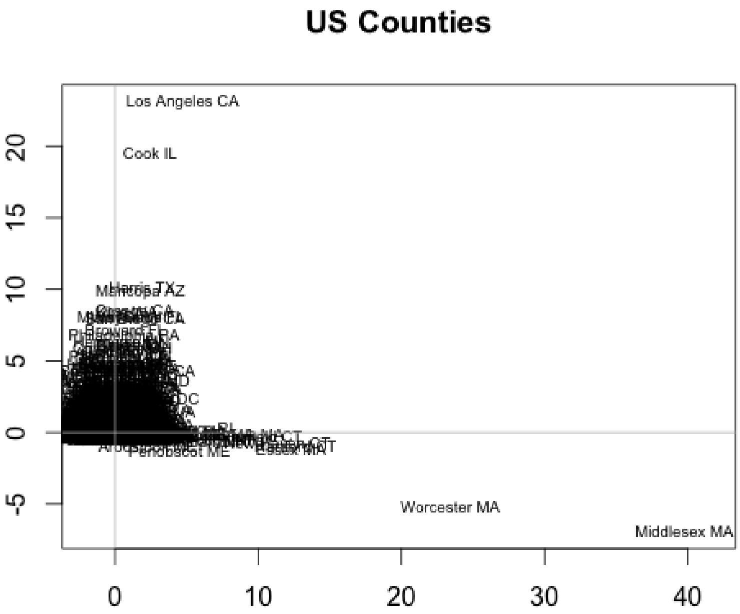
```
PsiB = diag((weights/sum(weights))^(-0.5)) %*%  eigB$vectors[,1:3] %*% diag(eigB$values[1:3]^(0.5))
rownames(PsiB) = Russet$pais
plot(PsiB,type="n")
text(PsiB,labels=Russet$pais,col=as.numeric(Russet$demo))
abline(h=0,v=0,col="gray")
```



```
loc <- cmdscale(D)
x <- loc[, 1];
y <- loc[, 2]
plot(x, y, type = "n", xlab = "", ylab = "", asp = 1, axes = TRUE, main = "distances between individuals")
text(x, y, Russet$pais, cex = 0.6,col=as.numeric(Russet$demo))
abline(h=0,v=0,col="gray")
```

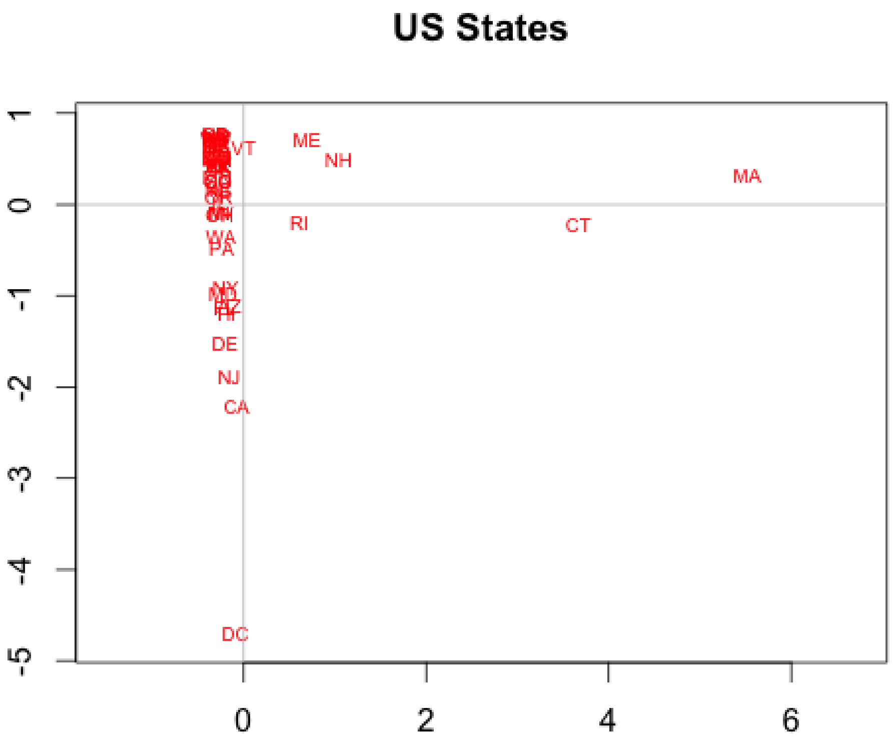


```
#Now I wanted to repeat on a data set I've composed of 2012 US presidential election data set by county.
edata <-read.csv("2012votes_health_merged_cleaned_final_version.csv", stringsAsFactors=TRUE)
dim(e.numeric)      #3113 counties with 63 variables (voting,socio-economic,public health)
nums <- sapply(edata, is.numeric)
e.numeric = edata[,nums]             #only use numeric variables for distance matrix input
e.numeric = e.numeric[,-c(1,2,3)]  #remove unnecessary ids
di <- dist(e.numeric)
loc2 <- cmdscale(di)
loc2 <- scale(loc2)
x.e <- loc2[, 1];  y.e <- loc2[, 2]
plot(x.e, y.e, type = "n", xlab = "", ylab = "", asp = 1, axes = TRUE, main = "US Counties")
text(x.e, y.e, paste(edata$county,edata$state.x), cex = 0.6,col="black")
abline(h=0,v=0,col="gray")
```



```
#This shows outlier counties pretty well! Los Angeles,CA , Cook,IL , Worcester,MA  and Middlesex,MA
#There are however 3113 counties which is too much to visualize so we'll lump together counties by state
```

```
library(psych)                          #for describeBy
e.numeric$state = edata$state.x         #add state info back for grouping purposes
states = describeBy(e.numeric[,-61],group=e.numeric$state)  #holds summaries of each state

hh = matrix(0,nrow=50,ncol=60)
colnames(hh) = rownames(tx)
rownames(hh) = names(states)
for(i in 1:50){ hh[i,] = round(states[[i]]$mean,4)}

di <- dist(hh)
loc2 <- cmdscale(di)
loc2 <- scale(loc2)
x.e <- loc2[, 1]
y.e <- as.vector(loc2[, 2])
state.abrvs = names(x.e)
x.e <- as.vector(x.e)
co = .1
plot(x.e, y.e, type = "n", xlab = "", ylab = "", asp = 1, axes = TRUE, main = "US States",
     xlim=c(min(x.e)-co,max(x.e)+co),ylim=c(min(y.e)-co,max(y.e)+co))
text(x.e, y.e, state.abrvs, cex = .6,col="red")
abline(h=0,v=0,col="gray")
```


US States
VT
ME
NH
MA
OR
RI
CT
WA
PA
NY
DE
NJ
CA
DC