Chapter 6 PLSC

Partial least square (PLS) methods (also sometimes called projection to latent structures) relate the information present in two data tables that collect measurements on the same set of observations. PLS methods proceed by deriving latent variables which are(optimal) linear combinations of the variables of a data table. When the goal is to find the shared information between two tables,the approach is equivalent to a correlation problem and the technique is then called partial least square correlation (PLSC) (also sometimes called PLS-SVD).

LETS START BY CLEARING THE ENVIRONMENT.

6.1 DATASET

For PLSC we will use the second part of the dataset

This data has same obserations as the data used in pca but variables are different.

Variables in this dataset are related to GDP population and other factors of the society.

For plsc we will divide the dataset into two parts population factors and GDP factors

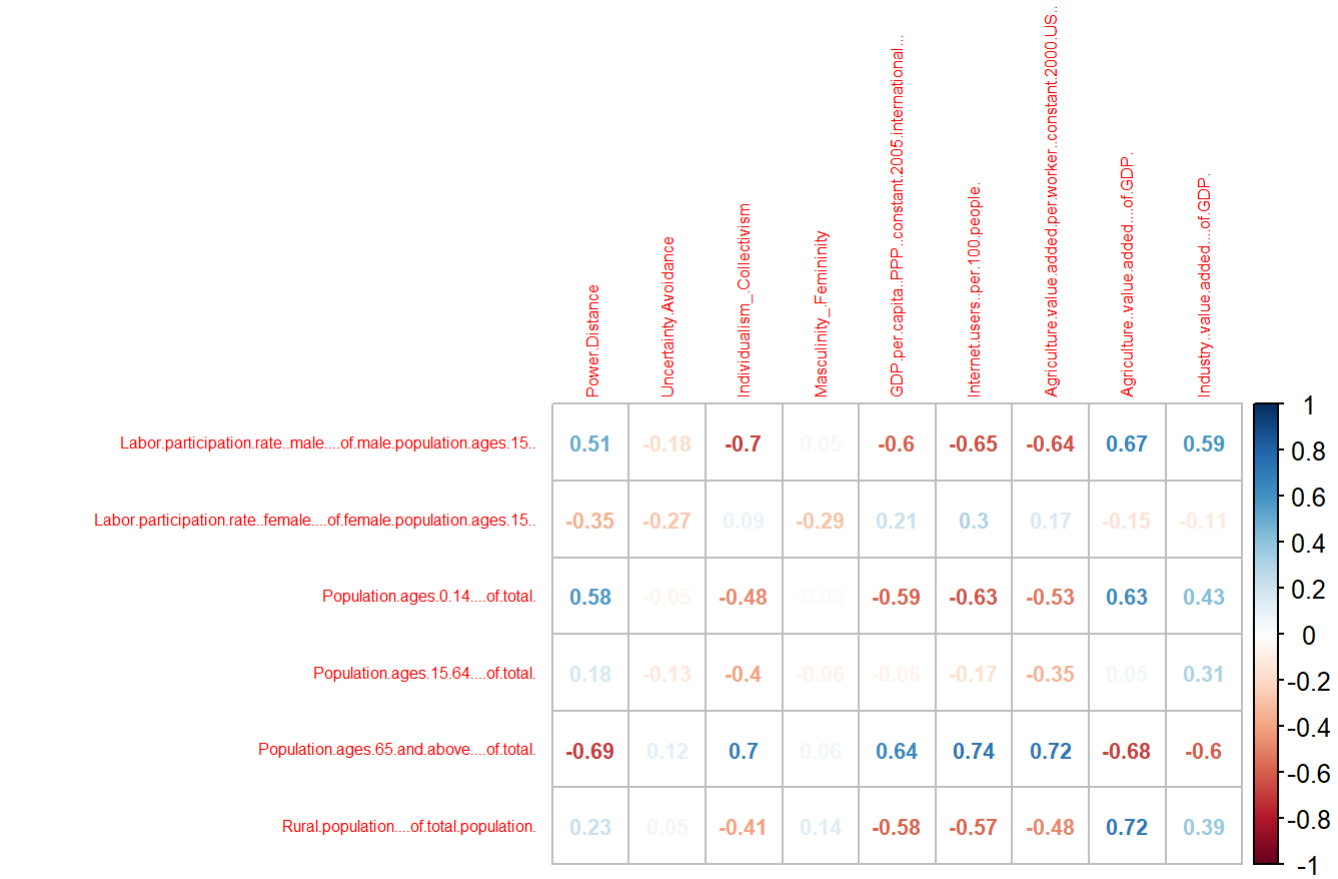
```
load("C:/Users/jeevan/Desktop/RM2/R-M/SmartphoneUsage (1).RData")
b=data.more
b["C"]=c("1","2","1","3","2","2","2","2","2","3","3","2","2","3","3","4","2","2","3","3","3
rawdata=b[-c(22),]
data1 <- rawdata[,4:12]
data2 <- rawdata[,14:19]
#DESIGN
data.design <- rawdata$C
data.design<-as.matrix(as.numeric(rawdata$C))
data.design.vec <- colSums(t(data.design)*c(1:ncol(data.design)))
group.key <- as.vector(colnames(data.design))
```

6.2 COR PLOT

```
library(corrplot)
```

```
XY.cor <- cor(data1,data2)
# Plot it with corrplotcolnames(a)=c('pd','apps','appsmon','FEUL','FEUM','FEULc','FEULc')
```

```
corrplot(t(XY.cor),method="number",tl.cex = 0.5,number.cex = 0.7)
```




```
cor.plot <- recordPlot()
```

6.3 SCREE PLOT

Dim 1 showed maximum variance

But for our analysis we will be using dim 1 and dim 2.

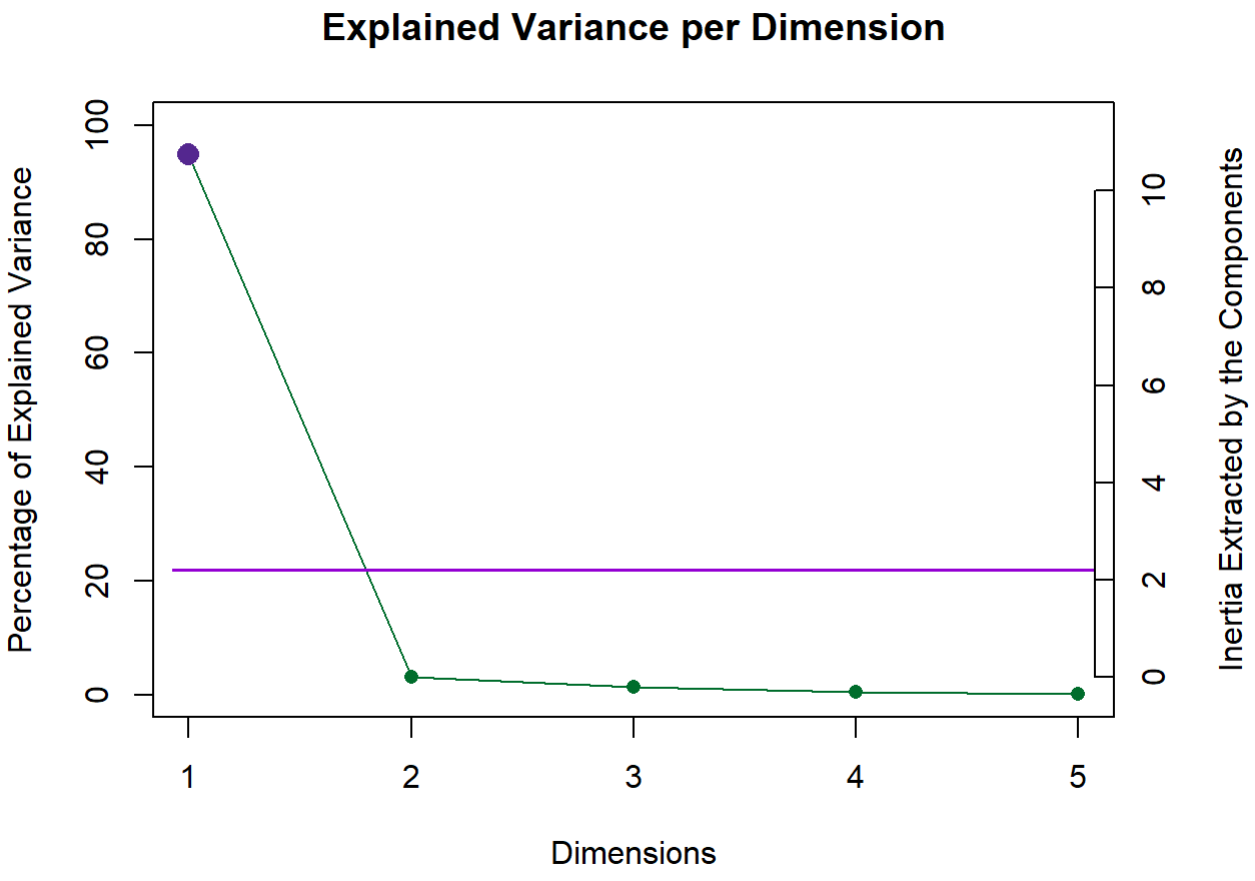
In plsc dimensions would be termed as latent variables.

```
#install.packages('tidyverse')
library(tidyverse)
library(ExPosition)
#install.packages('TExPosition') # if needed
library(TExPosition)
library(TInPosition)
library(PTCA4CATA)
# devtools::install_github('HerveAbdi/data4PCCAR')
library(data4PCCAR)
```

```
pls.res <- tepPLS(data1,data2, DESIGN = data.design, make_design_nominal = FALSE, graphs =
```

```
## [1] "Group Assignment Matrix is incorrect: too many items in the DESIGN matrix! Creating
```

```r
nL <- min(ncol(data1),ncol(data2))

resPerm4PLSC <- perm4PLSC(data1, # First Data matrix
                          data2, # Second Data matrix
                          nIter = 1000 # How mny iterations
                          )
# to see what results we have
#print(resPerm4PLSC)


my.scree<- PlotScree(ev=pls.res$TExPosition.Data$eigs,p.ev=resPerm4PLSC$pEigenvalues,plotKa
                     "darkviolet",lwd4Kaiser=1.5)
```



6.4 PROJECTIONS.

Grouping countries as per continents and taking the means with bootstrap intervals on latent variables

Note we have grouped our countries according to continents.

1=south america

2=europe

3=asia

4=north america

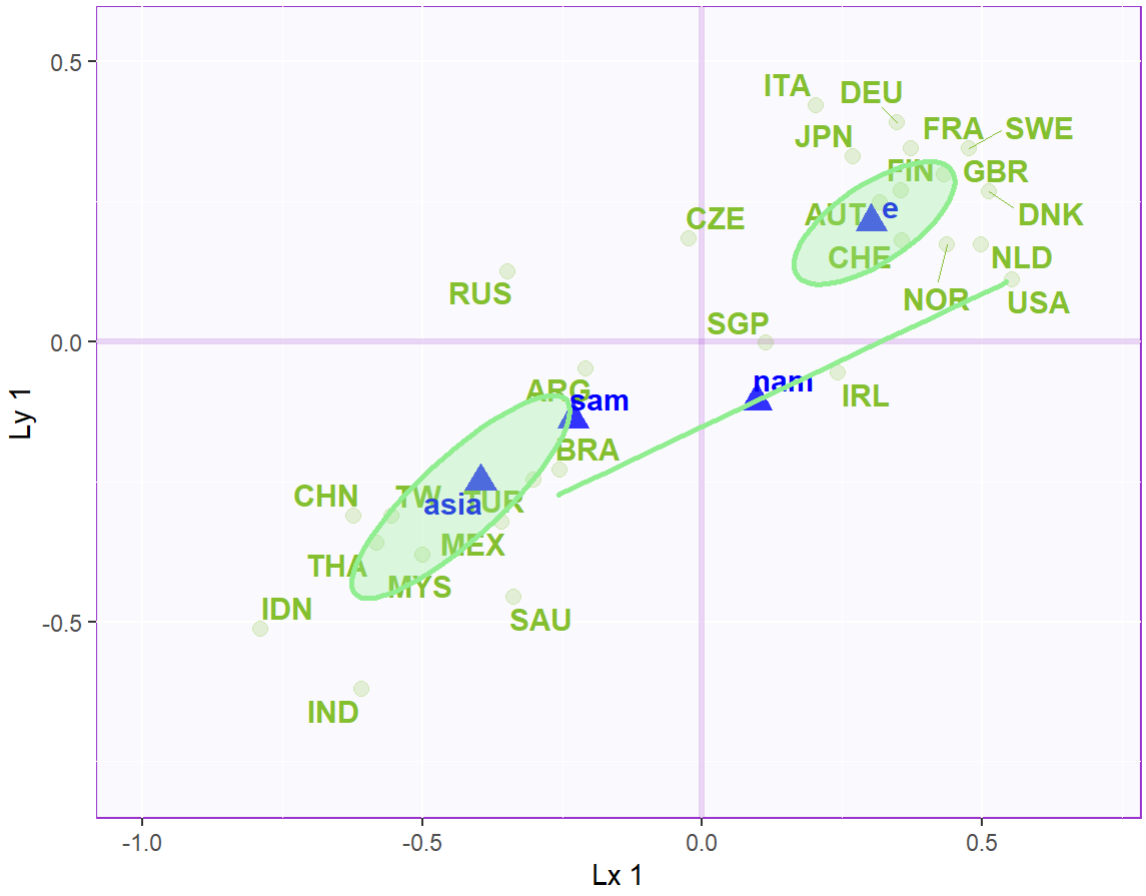
```r
# get colors for groups
uniquecol <- unique(pls.res$Plotting.Data$fii.col)
grpcol <- uniquecol
#rownames(grpcol) <- data.design.vec[rownames(uniquecol),]

# First, given how CreateFactorMap works, you need to create a matrix with observations on

# For the first plot, the first component of the latent variable of X is the x-axis, and t
latvar.1 <- cbind(pls.res$TExPosition.Data$lx[,1],pls.res$TExPosition.Data$ly[,1])
colnames(latvar.1) <- c("Lx 1", "Ly 1")

# compute means
lv.1.group <- getMeans(latvar.1,data.design.vec)
rownames(lv.1.group)=c('sam','e','asia','nam')

# get bootstrap intervals of groups
lv.1.group.boot <- Boot4Mean(latvar.1, data.design.vec)
colnames(lv.1.group.boot$BootCube) <- c("Lx 1", "Ly 1")

plot.lv1 <- createFactorMap(latvar.1,
                        col.points = pls.res$Plotting.Data$fii.col,
                        col.labels = pls.res$Plotting.Data$fii.col,
                        display.labels=TRUE,
                        alpha.points = 0.2
                        )

plot1.mean <- createFactorMap(lv.1.group,
                              col.points = 'blue',  #grpcol[rownames(lv.1.group),],
                              col.labels ='blue', #grpcol[rownames(lv.1.group),],

                              cex = 4,
                              pch = 17,
                              alpha.points = 0.8)

plot1.meanCI <- MakeCIEllipses(lv.1.group.boot$BootCube[,c(1:2),], # get the first two comp
```

```
                          col ="lightgreen", #grpcol[rownames(lv.1.group.boot$BootCube)

                          names.of.factors = c("Lx 1", "Ly 1")
                          )

plot1 <- plot.lv1$zeMap_background + plot.lv1$zeMap_dots+plot.lv1$zeMap_text + plot1.mean$z
plot1
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```

```
## Warning: Computation failed in `stat_ellipse()`:
## the leading minor of order 2 is not positive definite
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```


0.5
0.0
-0.5
Ly 1
-1.0
-0.5
0.0
0.5
Lx 1
ITA
DEU
JPN
FRA
SWE
FIN
GBR
CZE
AUT
e
DNK
CHE
NLD
RUS
NOR
USA
SGP
nam
IRL
ARG
sam
BRA
CHN
TW
asia
UR
MEX
THA
MYS
IDN
SAU
IND

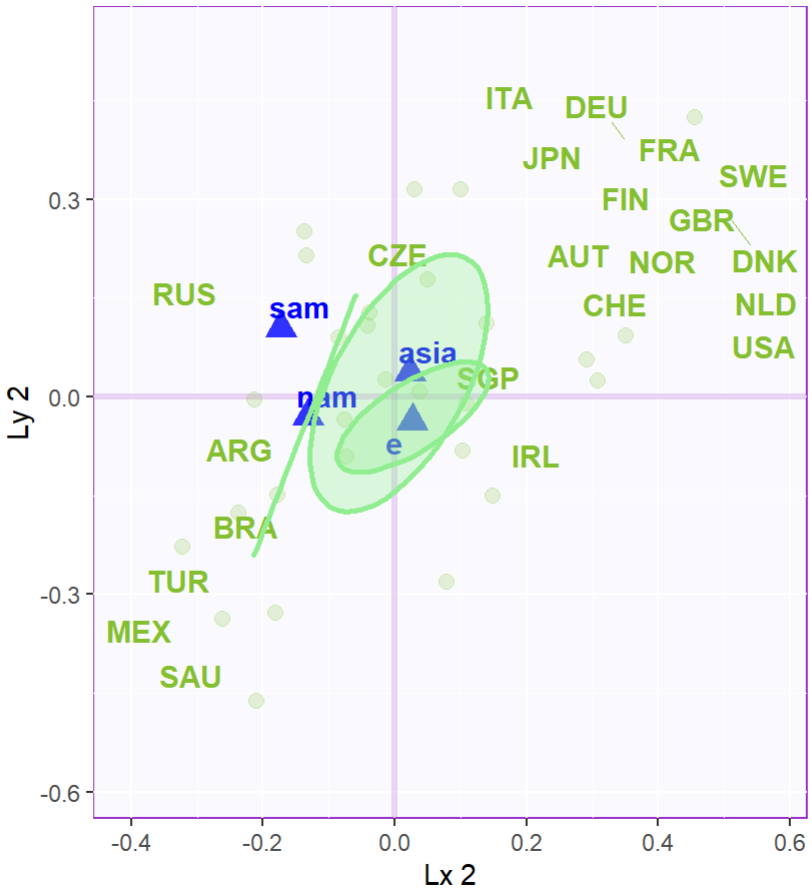
```r
latvar.2 <- cbind(pls.res$TExPosition.Data$lx[,2],pls.res$TExPosition.Data$ly[,2])
colnames(latvar.2) <- c("Lx 2", "Ly 2")

# compute means
lv.2.group <- getMeans(latvar.2,data.design)
rownames(lv.2.group)=c('sam','e','asia','nam')

# get bootstrap intervals of groups
lv.2.group.boot <- Boot4Mean(latvar.2, data.design)
colnames(lv.2.group.boot$BootCube) <- c("Lx 2", "Ly 2")
#Next, we can start plotting:

plot.lv2 <- createFactorMap(latvar.2,
                        col.points = pls.res$Plotting.Data$fii.col,
                        col.labels = pls.res$Plotting.Data$fii.col,
                        alpha.points = 0.2
                        )

plot2.mean <- createFactorMap(lv.2.group,
                              col.points = 'blue',#grpcol[rownames(lv.1.group),],
                              col.labels = 'blue',#grpcol[rownames(lv.1.group),],
                              cex=4,
                              text.cex = 4,
                              pch = 17,
                              alpha.points = 0.8)

plot2.meanCI <- MakeCIEllipses(lv.2.group.boot$BootCube[,c(1:2),], # get the first two comp
                              col ='lightgreen',# grpcol[rownames(lv.1.group.boot$BootCube)
                              names.of.factors = c("Lx 2", "Ly 2")
                              )

plot2 <- plot.lv2$zeMap_background + plot.lv2$zeMap_dots +plot.lv1$zeMap_text+ plot2.mean$z
plot2
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```

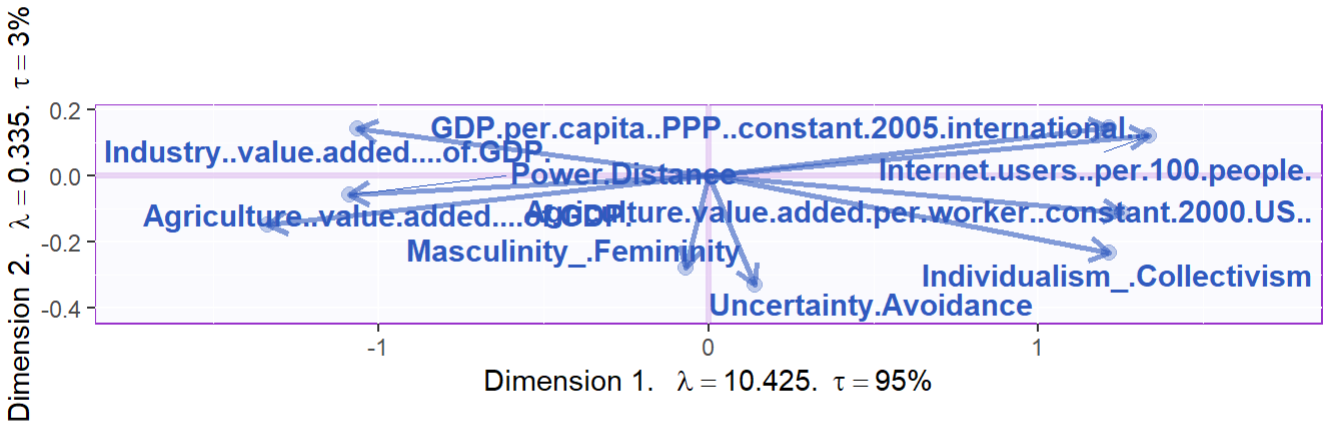
```
## Warning: Computation failed in `stat_ellipse()`:
## the leading minor of order 2 is not positive definite
```

```
## Warning in MASS::cov.trob(data[, vars]): Probable convergence failure
```

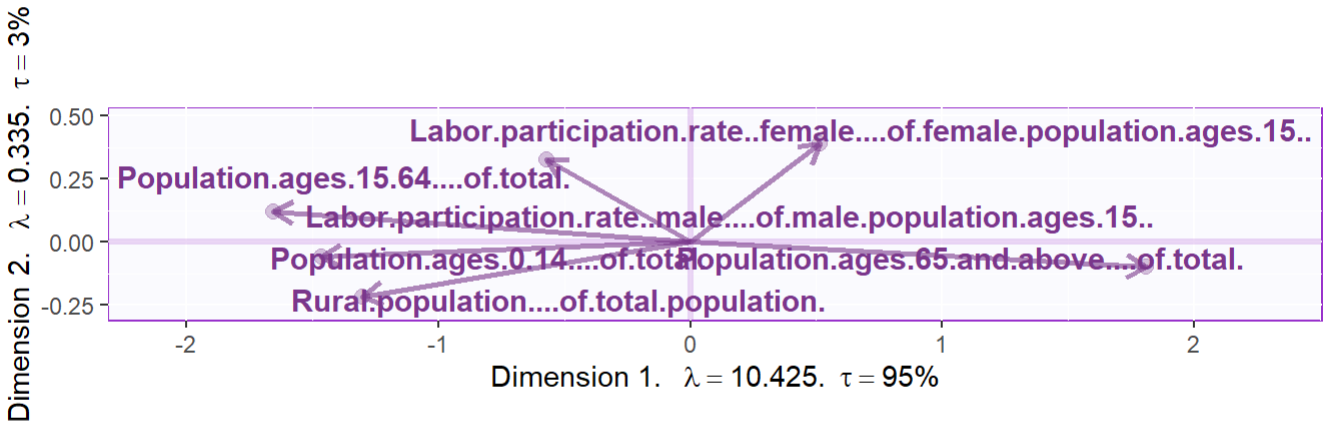
```
## Warning: Removed 6 rows containing missing values (geom_text_repel).
```



```r
col4X <- prettyGraphsColorSelection(n.colors = ncol(data1),
                                    starting.color = 42)

col4Y <- prettyGraphsColorSelection(n.colors = ncol(data2),
                                    starting.color = 13)
col4Var = c(col4X,col4Y)
#_____________________________________________________________________
Fj <- pls.res$TExPosition.Data$fj
Fi <- pls.res$TExPosition.Data$fi
baseMap.i <- PTCA4CATA::createFactorMap(Fi,
                                        col.points   = pls.res$Plotting.Data$fi.col,
                                        alpha.points =  .3,
                                        col.labels   = pls.res$Plotting.Data$fi.col)
# arrows
zeArrows <- addArrows(Fi, color = pls.res$Plotting.Data$fi.col)
# A graph for the J-set
# A graph for the J-set
label4Map <- createxyLabels.gen(1,2,
                                lambda = pls.res$TExPosition.Data$eigs,
                                tau = pls.res$TExPosition.Data$t)

b001.aggMap.i <- baseMap.i$zeMap_background + # background layer
  baseMap.i$zeMap_dots + baseMap.i$zeMap_text +  # dots & labels
  label4Map
b002.aggMap.i <- b001.aggMap.i + zeArrows
# We print this Map with the following code

b002.aggMap.i
```


Dimension 2. λ = 0.335. τ = 3%
0.2
0.0
-0.2
-0.4
GDP.per.capita..PPP..constant.2005.international.
Industry..value.added....of.GDP.
Power.Distance
Internet.users..per.100.people.
Agriculture..value.added....of.GDP.
Agriculture.value.added.per.worker..constant.2000.US..
Masculinity_.Femininity
Individualism_.Collectivism
Uncertainty.Avoidance
-1
0
1
Dimension 1. λ = 10.425. τ = 95%

```r
Fj <- pls.res$TExPosition.Data$fj

baseMap.j <- PTCA4CATA::createFactorMap(Fj,
                                        col.points   = pls.res$Plotting.Data$fj.col,
                                        alpha.points =  .3,
                                        col.labels   = pls.res$Plotting.Data$fj.col)
# arrows
zeArrows <- addArrows(Fj, color = pls.res$Plotting.Data$fj.col)
# A graph for the J-set
# A graph for the J-set
label4Map <- createxyLabels.gen(1,2,
                                lambda = pls.res$TExPosition.Data$eigs,
                                tau = pls.res$TExPosition.Data$t)

b001.aggMap.j <- baseMap.j$zeMap_background + # background layer
  baseMap.j$zeMap_dots + baseMap.j$zeMap_text +  # dots & labels
  label4Map
b002.aggMap.j <- b001.aggMap.j + zeArrows
# We print this Map with the following code

b002.aggMap.j
```



6.5 ANALYSIS/HYPOTHESIS

From the plot lx1 and ly1 it is evident that south america and asia are closer and europe and north america are closer.

This implies that rich countries have similar economical and population factors. and developing countries have similar economical and population factors.

From the plot lx2 and ly2 it is evident that all continents are pretty close to each other however we cannot conclude anything as latent variable 2 is not significant.

From PCA and plsc analysis we can have an hypothesis that smartphones sales is mainly related to the population factors and economical factors.

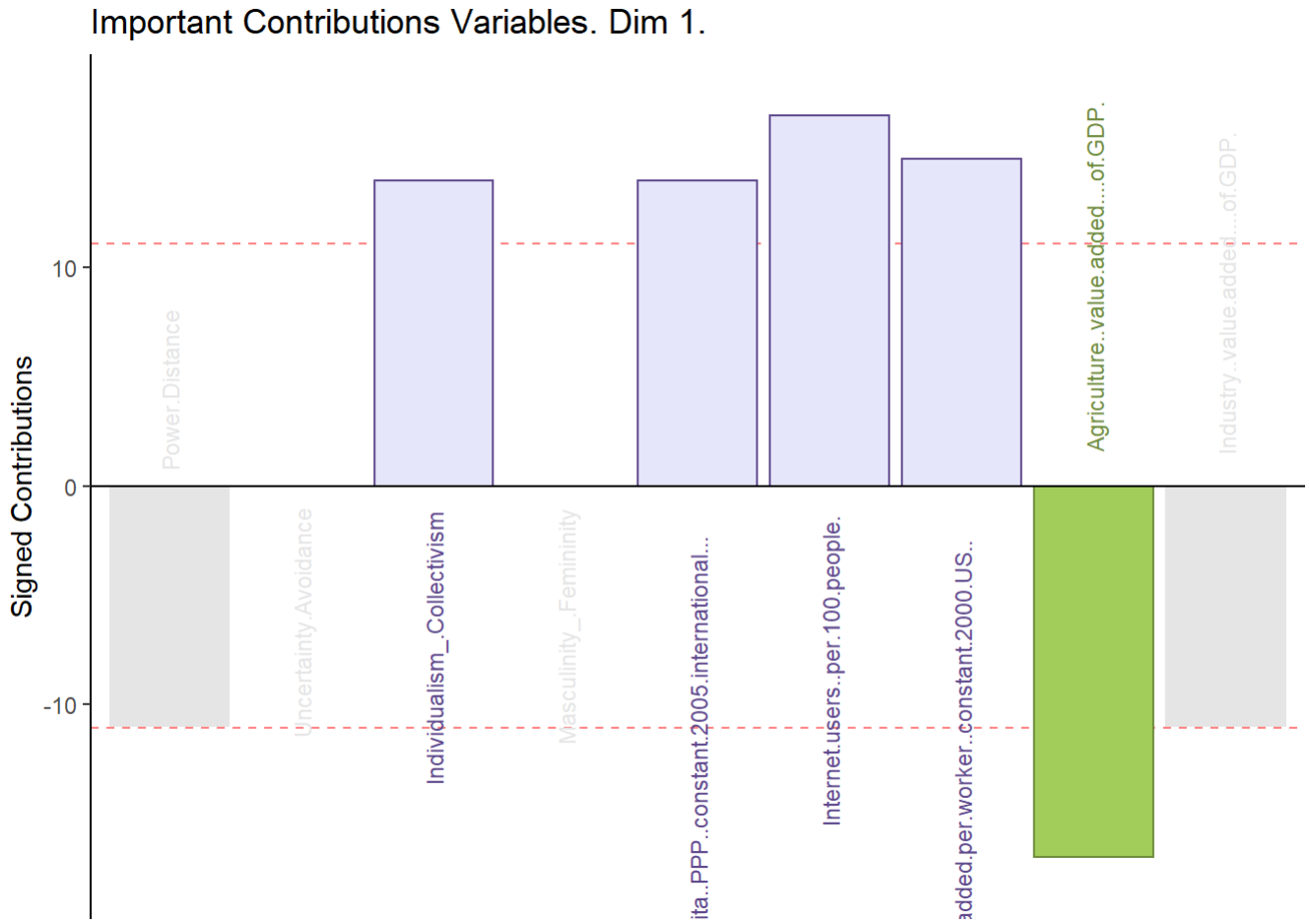
6.6 CONTRIBUTIONS

This plot will show us how variables from both the dataset contribute to the latent variables.

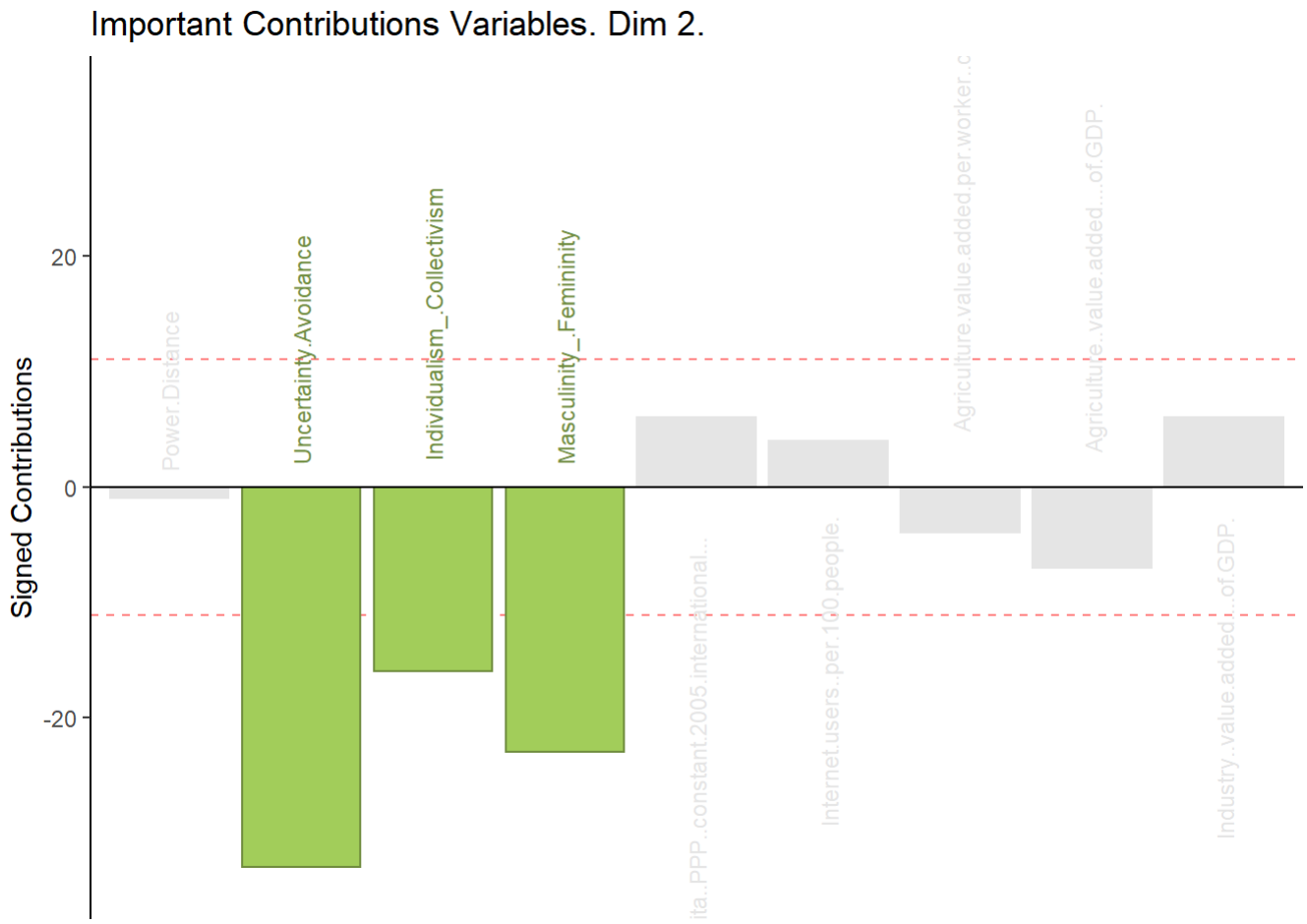
Variables from both the dataset contribute more for the latent variable x (dim1).

```r
ctri <- pls.res$TExPosition.Data$ci
signed.ctri <- ctri * sign(Fi)
# BR1
c001.plotCtri.1 <- PrettyBarPlot2(
                        bootratio = round(100*signed.ctri[,1]),
                        threshold = 100 / nrow(signed.ctri),
                        ylim = NULL,
                        color4bar = NULL ,#gplots::col2hex(col4Var),
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Important Contributions Variables. Dim 1.',
                        ylab = "Signed Contributions")



c001.plotCtri.2 <- PrettyBarPlot2(
                        bootratio = round(100*signed.ctri[,2]),
                        threshold = 100 / nrow(signed.ctri),
                        ylim = NULL,
                        color4bar = NULL ,#gplots::col2hex(col4Var),
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Important Contributions Variables. Dim 2.',
                        ylab = "Signed Contributions")



ctrj <- pls.res$TExPosition.Data$cj
signed.ctrj <- ctrj * sign(Fj)
# BR1
c001.plotCtrj.1 <- PrettyBarPlot2(
                        bootratio = round(100*signed.ctrj[,1]),
                        threshold = 100 / nrow(signed.ctrj),
```

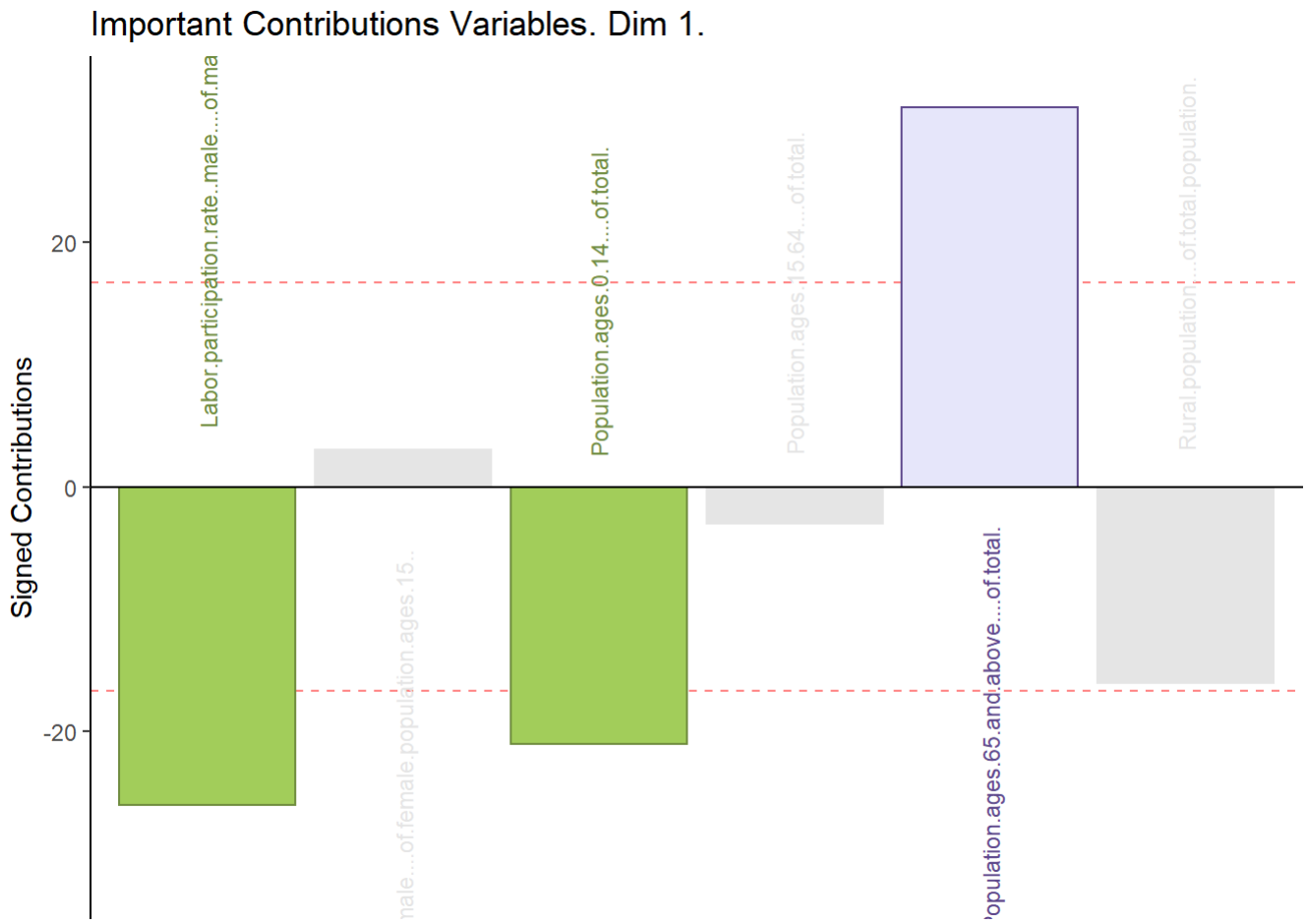
```
                        ylim = NULL,
                        color4bar = NULL ,#gplots::col2hex(col4Var),
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Important Contributions Variables. Dim 1.',
                        ylab = "Signed Contributions")


c001.plotCtrj.2<- PrettyBarPlot2(
                        bootratio = round(100*signed.ctrj[,2]),
                        threshold = 100 / nrow(signed.ctrj),
                        ylim = NULL,
                        color4bar = NULL ,#gplots::col2hex(col4Var),
                        color4ns = "gray75",
                        plotnames = TRUE,
                        main = 'Important Contributions Variables. Dim 2.',
                        ylab = "Signed Contributions")



c001.plotCtri.1
```



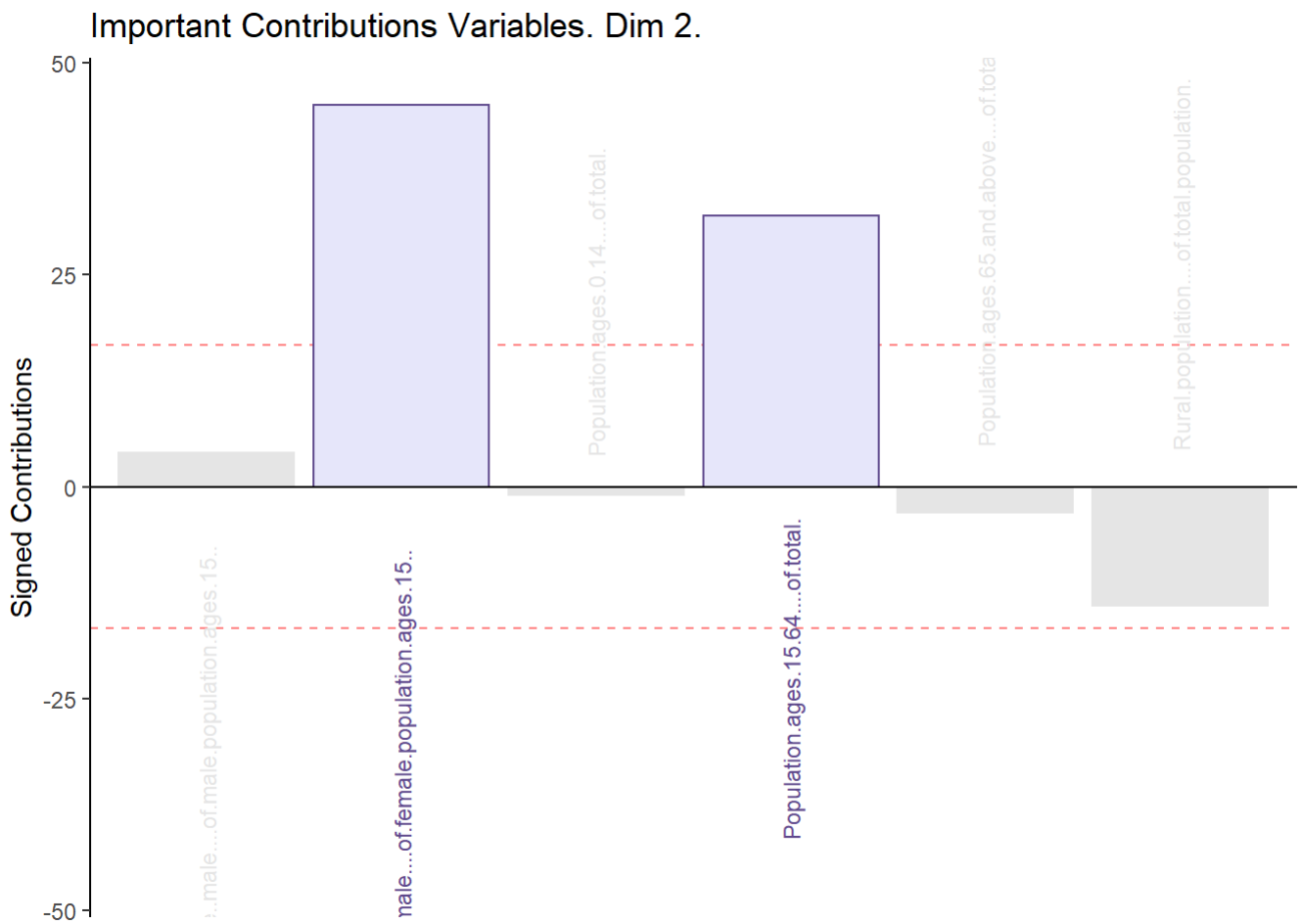

```
c001.plotCtri.2
```




```
c001.plotCtrj.1
```




```
c001.plotCtrj.2
```




```
library(grid)
library(gridExtra)
library(gridGraphics)
library(ggplot2)
library(ggplotify)
```

6.7 BOOTSTRAP

Checking the reliability of our interpreations and hypothesis with bootstrap.

```
resBoot4PLSC <- Boot4PLSC(data1, # First Data matrix
                          data2, # Second Data matrix
                          nIter = 1000, # How many iterations
                        Fi = pls.res$TExPosition.Data$fi,
                        Fj = pls.res$TExPosition.Data$fj,
                        nf2keep = 3,
                        critical.value = 2,
                        # To be implemented later
                        # has no effect currently
                        alphaLevel = .05)
#
# to see what results we have
resBoot4PLSC


## ------------------------------------------------------------------------
##  Bootstraped Factor Scores (BFS) and Bootstrap Ratios  (BR)
##  for the I and J-sets of a PLSC (obtained from multinomial resampling of X & Y)
## ------------------------------------------------------------------------
## $ bootstrapBrick.i          an I*L*nIter Brick of BFSs  for the I-Set
## $ bootRatios.i              an I*L matrix of BRs for the I-Set
## $ bootRatiosSignificant.i   an I*L logical matrix for significance of the I-Set
## $ bootstrapBrick.j          a  J*L*nIter Brick of BFSs  for the J-Set
## $ bootRatios.j              a  J*L matrix of BRs for the J-Set
## $ bootRatiosSignificant.j   a  J*L logical matrix for significance of the J-Set
## ------------------------------------------------------------------------
```

```r
BR.I<-resBoot4PLSC$bootRatios.i
BR.J<-resBoot4PLSC$bootRatios.j

laDim=1

# Plot the bootstrap ratios for Dimension 1
ba001.BR1.I <- PrettyBarPlot2(BR.I[,laDim],
                         threshold = 2,
                         font.size = 3,
                      color4bar = gplots::col2hex(pls.res$Plotting.Data$fi.col), # we need he
                     ylab = 'Bootstrap ratios'
                     #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle(paste0('Component ', laDim), subtitle = 'Table 1')



ba002.BR1.J <- PrettyBarPlot2(BR.J[,laDim],
                         threshold = 2,
                         font.size = 3,
                      color4bar = gplots::col2hex(pls.res$Plotting.Data$fj.col), # we need he
                     ylab = 'Bootstrap ratios'
                     #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle("", subtitle = 'Table 2')

# Plot the bootstrap ratios for Dimension 2
laDim = 2
ba003.BR2.I <- PrettyBarPlot2(BR.I[,laDim],
                         threshold = 2,
                         font.size = 3,
                      color4bar = gplots::col2hex(pls.res$Plotting.Data$fi.col), # we need he
                     ylab = 'Bootstrap ratios'
                     #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle(paste0('Component ', laDim), subtitle = 'Table 1')

ba004.BR2.J <- PrettyBarPlot2(BR.J[,laDim],
```

```
                        threshold = 2,
                        font.size = 3,
                     color4bar = gplots::col2hex(pls.res$Plotting.Data$fj.col), # we need hex
                    ylab = 'Bootstrap ratios'
                    #ylim = c(1.2*min(BR[,laDim]), 1.2*max(BR[,laDim]))
) + ggtitle("", subtitle = 'Table 2')
```

```
library(grid)
library(gridExtra)
library(gridGraphics)
library(ggplot2)
library(ggplotify)
grid.arrange(
    as.grob(ba001.BR1.I),as.grob(ba002.BR1.J),as.grob(ba003.BR2.I),as.grob(ba004.BR2.J),
    ncol = 2,nrow = 2,
    top = textGrob("Bootstrap ratios", gp = gpar(fontsize = 18, font = 3))
  )
```

Bootstrap ratios

Component 1

Table 1

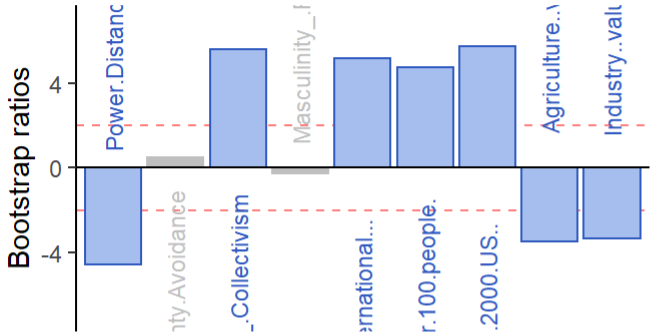
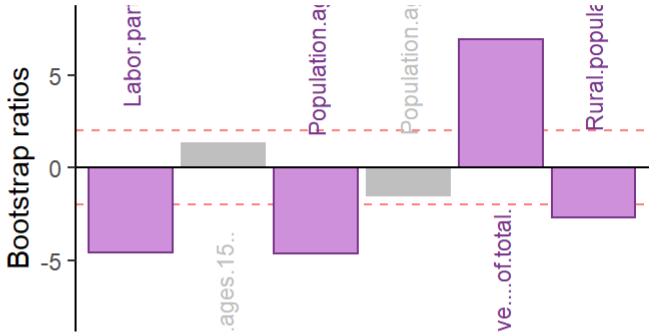


Table 2



Component 2

Table 1

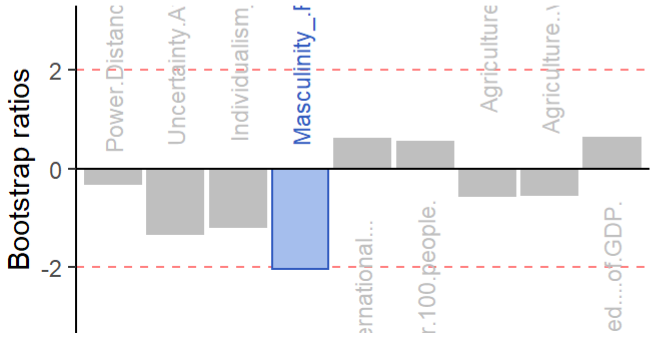


Table 2