All Countries Environmental Data

## Method: MCA

Multiple correspondence analysis (MCA) is an extension of correspondence analysis(CA) which allows one to analyze the pattern of relationships of several categorical dependent variables. As such, it can also be seen as a generalization of principal component analysis when the variables to be analyzed are categorical instead of quantitative. Because MCA has been (re)discovered many times, equivalent methods are known under several different names such as optimal scaling, optimal or appropriate scoring, dual scaling, homogeneity analysis,scalogram analysis, and quantiﬁcation method.

**Interpreting MCA** Multiple correspondence analysis locates all the categories in a Euclidean space.

* The first two dimensions of this space are plotted to examine the associations among the categories.
* The top-right quadrant of the plot shows the categories.
* The bottom-left quadrant shows the association.
* This interpretation is based on points found in approximately the same direction from the origin and in approximately the same region of the space. Distances between points do not have a straightforward interpretation.

## Dataset

* Data: Measurements of environment conditions in Countries
* Rows: There are 137 observations, 1 for each country.
* Columns: Total 29 variables
* Qualitative: Country (nominal), Happiness (Ordinal).
* Quantitative: Aspect, Slope Crop Land, Tree Canopy Wind Cloud & Multiple variables for Temp & Rain
* Structure of Data

str(country\_env\_df)

## 'data.frame': 137 obs. of 29 variables:  
## $ Country : Factor w/ 137 levels "Afghanistan",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ Happiness\_Rank : Ord.factor w/ 3 levels "VH"<"H"<"U": 3 2 2 3 1 3 1 1 2 2 ...  
## $ accessibility\_to\_cities: num 317.7 73.8 1212.8 378.2 209.2 ...  
## $ elevation : num 1832 652 557 1061 683 ...  
## $ aspect : num 201 192 185 174 145 ...  
## $ slope : num 1.516 1.89 0.171 0.193 0.624 ...  
## $ cropland\_cover : num 9.51 23.35 3.69 2.79 21.96 ...  
## $ tree\_canopy\_cover : num 0.375 12.805 0.177 19.87 8.834 ...  
## $ isothermality : num 35.9 33.2 40.3 64.3 49.9 ...  
## $ rain\_coldestQuart : num 128.72 392.51 25.29 8.05 79.09 ...  
## $ rain\_driestMonth : num 1.722 40.088 0.935 0.26 17.183 ...  
## $ rain\_driestQuart : num 8.3 138.15 6.09 4.43 60.49 ...  
## $ rain\_mean\_annual : num 311.3 1151.1 79.5 1023.4 539.9 ...  
## $ rain\_seasonailty : num 91.6 38.5 67.1 91.5 48.3 ...  
## $ rain\_warmestQuart : num 12.69 138.33 9.51 318.54 183.14 ...  
## $ rain\_wettestMonth : num 67.8 159 13.4 202.2 79.2 ...  
## $ rain\_wettestQuart : num 175.8 435.9 33.3 524.3 211.7 ...  
## $ temp\_annual\_range : num 40.3 27.1 36.5 21.5 26.8 ...  
## $ temp\_coldestQuart : num -0.261 3.58 13.152 18.794 8.024 ...  
## $ temp\_diurnal\_range : num 14.72 9.11 14.87 13.85 13.46 ...  
## $ temp\_driestQuart : num 21.1 19.6 26.9 18.9 11.1 ...  
## $ temp\_max\_warmestMonth : num 32 26.3 41.5 31 28.2 ...  
## $ temp\_mean\_annual : num 11.5 11.5 23 21.6 14.2 ...  
## $ temp\_min\_coldestMonth : num -8.312 -0.806 5.058 9.549 1.443 ...  
## $ temp\_seasonality : num 88.2 62.7 75.1 18.5 47.6 ...  
## $ temp\_warmestQuart : num 22.7 19.6 32.5 23.3 20.2 ...  
## $ temp\_wettestQuart : num 3.95 5.27 20.81 22.76 16.48 ...  
## $ wind : num 3.43 2.47 4.03 2.16 4.27 ...  
## $ cloudiness : num 114.2 181.1 90.7 187.5 159 ...

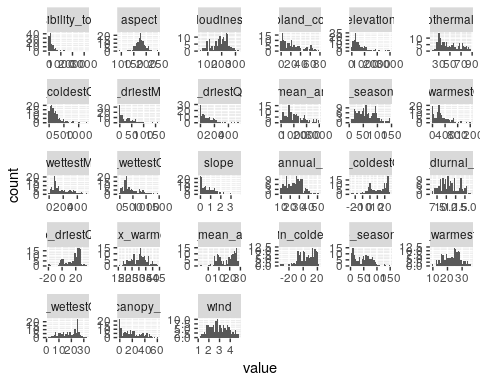
* Research Question

How do the 137 countries differ on these variables?

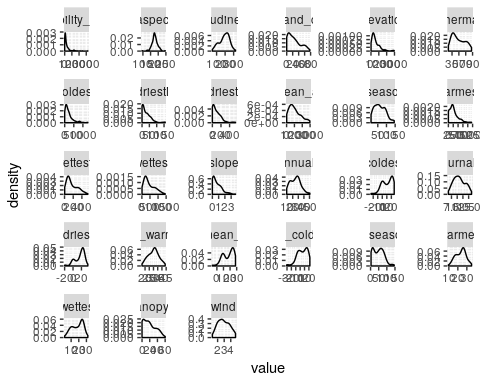
## Analysis

Let’s observe the distribution of each variables to get an intuition of how we can bin these variables. It’s important to have nearly equal number of observations in the each bin and to try to cut the variables in a way to so that each new binned distribution is nearly Gaussian. We can also verify that our binning is appropiate by calculating Spearman Correlation for each of original variable and binned variable, the correlation coefficient should be close to 1.

### Histogram



### Density plot



### Binning

We will use cut method with 4 quartile intervals for each variable to get binned version of each variable.

cut\_r <- function(x, b = 4, label=c(1,2,3,4) ) {  
 c <- cut(x,breaks = b, labels = label)  
 #h <- hclust(dist(sort(country\_env\_df\_for\_pca[,1])),"median")  
 #c <- cutree(h, k = b)  
 return (factor(c))  
}  
  
country\_env\_df\_for\_mca = lapply(country\_env\_df\_for\_pca, cut\_r, )  
country\_env\_df\_for\_mca <- as.data.frame(country\_env\_df\_for\_mca)  
str(country\_env\_df\_for\_mca)

## 'data.frame': 137 obs. of 27 variables:  
## $ accessibility\_to\_cities: Factor w/ 4 levels "1","2","3","4": 1 1 2 1 1 1 2 1 1 1 ...  
## $ elevation : Factor w/ 4 levels "1","2","3","4": 3 1 1 2 1 3 1 2 1 1 ...  
## $ aspect : Factor w/ 4 levels "1","2","3","4": 3 3 3 3 2 3 3 2 1 2 ...  
## $ slope : Factor w/ 4 levels "1","2","3","4": 2 3 1 1 1 3 1 2 2 1 ...  
## $ cropland\_cover : Factor w/ 4 levels "1","2","3","4": 1 2 1 1 2 2 1 2 2 4 ...  
## $ tree\_canopy\_cover : Factor w/ 4 levels "1","2","3","4": 1 1 1 2 1 1 1 3 1 2 ...  
## $ isothermality : Factor w/ 4 levels "1","2","3","4": 1 1 2 3 2 1 2 1 1 2 ...  
## $ rain\_coldestQuart : Factor w/ 4 levels "1","2","3","4": 1 2 1 1 1 1 1 1 1 1 ...  
## $ rain\_driestMonth : Factor w/ 4 levels "1","2","3","4": 1 2 1 1 1 1 1 2 1 1 ...  
## $ rain\_driestQuart : Factor w/ 4 levels "1","2","3","4": 1 2 1 1 1 1 1 2 1 1 ...  
## $ rain\_mean\_annual : Factor w/ 4 levels "1","2","3","4": 1 2 1 2 1 1 1 2 1 3 ...  
## $ rain\_seasonailty : Factor w/ 4 levels "1","2","3","4": 3 1 2 3 2 2 2 1 1 3 ...  
## $ rain\_warmestQuart : Factor w/ 4 levels "1","2","3","4": 1 1 1 2 1 1 1 2 1 4 ...  
## $ rain\_wettestMonth : Factor w/ 4 levels "1","2","3","4": 1 2 1 2 1 1 1 1 1 4 ...  
## $ rain\_wettestQuart : Factor w/ 4 levels "1","2","3","4": 1 2 1 2 1 1 1 2 1 4 ...  
## $ temp\_annual\_range : Factor w/ 4 levels "1","2","3","4": 3 2 3 2 2 3 2 2 3 2 ...  
## $ temp\_coldestQuart : Factor w/ 4 levels "1","2","3","4": 2 3 3 4 3 2 4 2 3 4 ...  
## $ temp\_diurnal\_range : Factor w/ 4 levels "1","2","3","4": 4 2 4 4 3 3 4 2 2 2 ...  
## $ temp\_driestQuart : Factor w/ 4 levels "1","2","3","4": 3 3 4 3 3 2 3 2 3 3 ...  
## $ temp\_max\_warmestMonth : Factor w/ 4 levels "1","2","3","4": 3 2 4 3 2 2 3 1 2 3 ...  
## $ temp\_mean\_annual : Factor w/ 4 levels "1","2","3","4": 3 3 4 4 3 2 4 2 3 4 ...  
## $ temp\_min\_coldestMonth : Factor w/ 4 levels "1","2","3","4": 2 3 3 4 3 2 3 2 3 4 ...  
## $ temp\_seasonality : Factor w/ 4 levels "1","2","3","4": 3 2 3 1 2 3 2 2 3 1 ...  
## $ temp\_warmestQuart : Factor w/ 4 levels "1","2","3","4": 3 2 4 3 2 2 3 1 2 3 ...  
## $ temp\_wettestQuart : Factor w/ 4 levels "1","2","3","4": 1 1 3 3 2 2 4 2 2 4 ...  
## $ wind : Factor w/ 4 levels "1","2","3","4": 3 2 4 2 4 1 4 2 2 2 ...  
## $ cloudiness : Factor w/ 4 levels "1","2","3","4": 1 2 1 3 2 3 2 3 3 3 ...

### Spearman Correlation

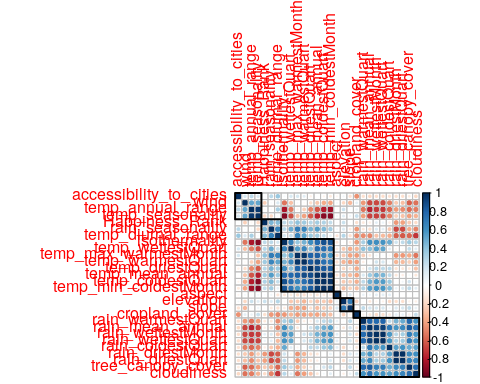
for (i in 1:ncol(country\_env\_df\_for\_mca)){  
 print(paste(colnames(country\_env\_df\_for\_mca)[i], "--Correlation--", cor(as.integer(country\_env\_df\_for\_mca[,i]),as.integer(country\_env\_df\_for\_pca[,i]),method = "spearman")))  
}

## [1] "accessibility\_to\_cities --Correlation-- 0.525401822909277"  
## [1] "elevation --Correlation-- 0.790183929423767"  
## [1] "aspect --Correlation-- 0.85932453017731"  
## [1] "slope --Correlation-- 0.931258681983175"  
## [1] "cropland\_cover --Correlation-- 0.926727929119737"  
## [1] "tree\_canopy\_cover --Correlation-- 0.938012761345693"  
## [1] "isothermality --Correlation-- 0.962499862945234"  
## [1] "rain\_coldestQuart --Correlation-- 0.711666324484794"  
## [1] "rain\_driestMonth --Correlation-- 0.784467426611677"  
## [1] "rain\_driestQuart --Correlation-- 0.78956885174067"  
## [1] "rain\_mean\_annual --Correlation-- 0.936865893295628"  
## [1] "rain\_seasonailty --Correlation-- 0.952266060848619"  
## [1] "rain\_warmestQuart --Correlation-- 0.823015920402402"  
## [1] "rain\_wettestMonth --Correlation-- 0.908913074326508"  
## [1] "rain\_wettestQuart --Correlation-- 0.913890327515428"  
## [1] "temp\_annual\_range --Correlation-- 0.946095668473989"  
## [1] "temp\_coldestQuart --Correlation-- 0.926060263980741"  
## [1] "temp\_diurnal\_range --Correlation-- 0.954709642185881"  
## [1] "temp\_driestQuart --Correlation-- 0.934067623393866"  
## [1] "temp\_max\_warmestMonth --Correlation-- 0.945837071960447"  
## [1] "temp\_mean\_annual --Correlation-- 0.919155959215819"  
## [1] "temp\_min\_coldestMonth --Correlation-- 0.935315491499545"  
## [1] "temp\_seasonality --Correlation-- 0.936582400632849"  
## [1] "temp\_warmestQuart --Correlation-- 0.932630129083355"  
## [1] "temp\_wettestQuart --Correlation-- 0.958719803820619"  
## [1] "wind --Correlation-- 0.949786797554234"  
## [1] "cloudiness --Correlation-- 0.93065137395011"

### Correlation Plot

Visually analyze multicollinearity in the system.

corr\_result = cor(country\_env\_df\_for\_corr)  
corrplot(corr\_result,order = 'hclust', addrect = 7)



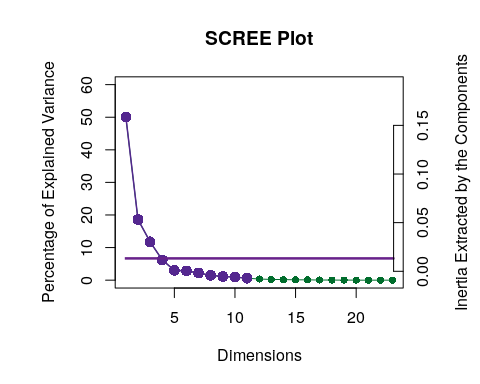
## Identify Latent Components - MCA (With Inference)

### Scree Plot

Gives amount of information explained by corresponding component. Gives an intuition to decide which components best represent data in order to answer the research question.

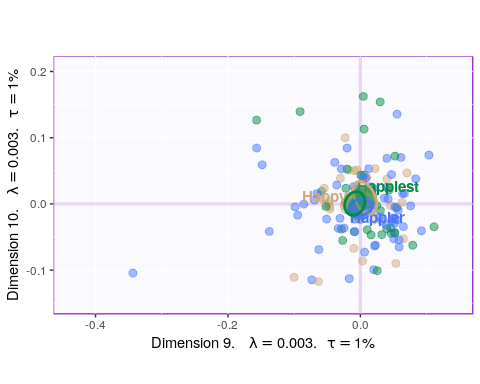
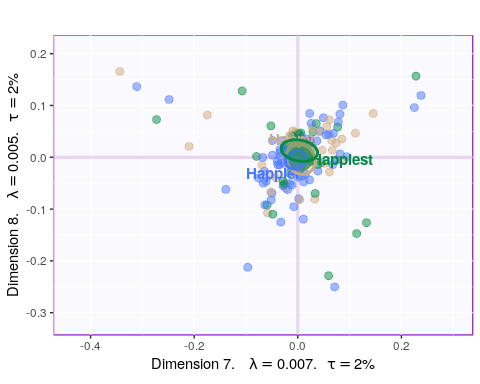
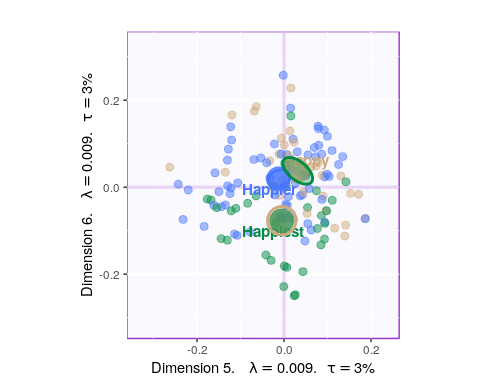
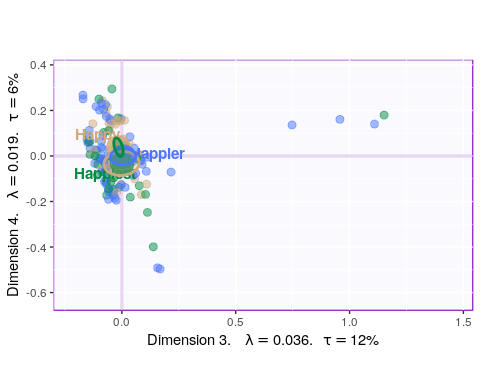
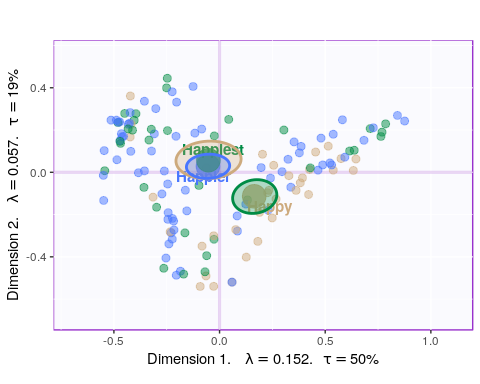
P.S. The most contribution component may not always be most useful for a given research question.

PTCA4CATA::PlotScree(ev = country\_env\_mca$ExPosition.Data$eigs,  
 p.ev = country\_env\_mca\_inf$Inference.Data$components$p.vals,  
 title = 'SCREE Plot',  
 plotKaiser = TRUE  
)

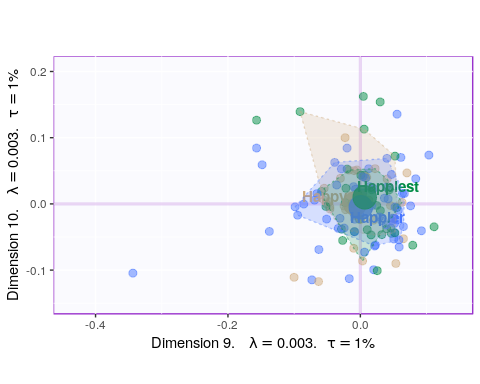
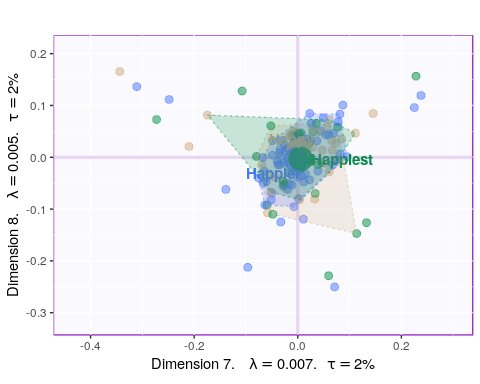
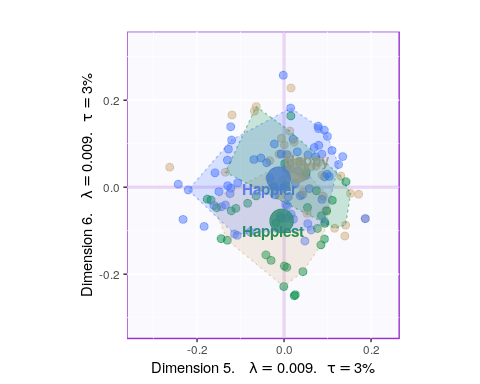
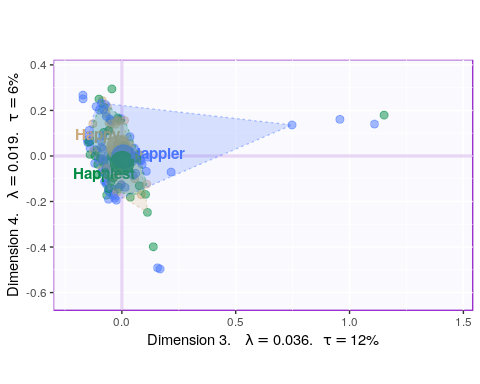
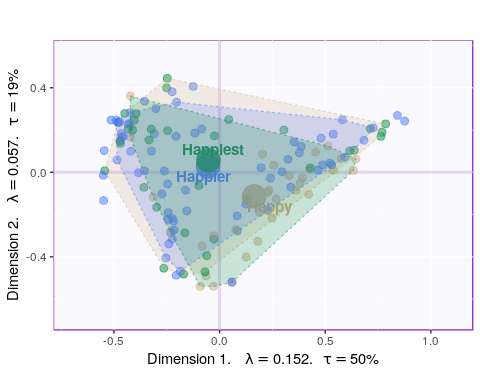


### Factor Scores

Lets visualize happiness categories for components 1-10, to make a decision (visually) on the most important components.

**With Confidence Interval** 

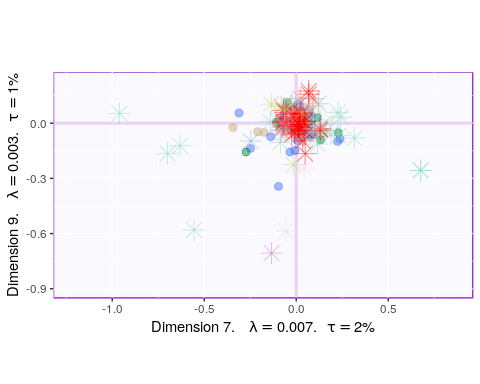
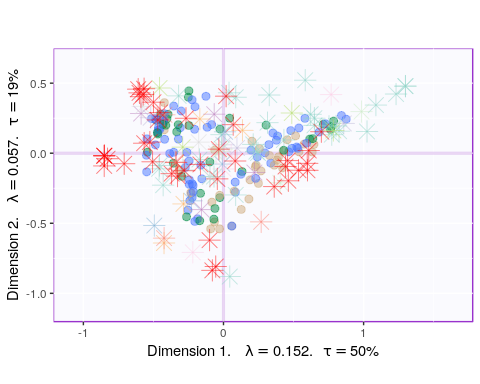
**With Tolerance Interval**



We will proceed by making Symmetric and Asymmetric plots for Components 1,2,7,9.

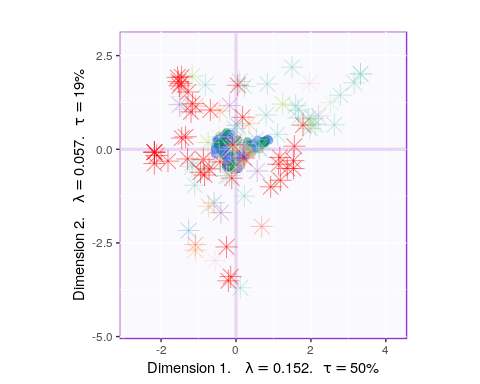
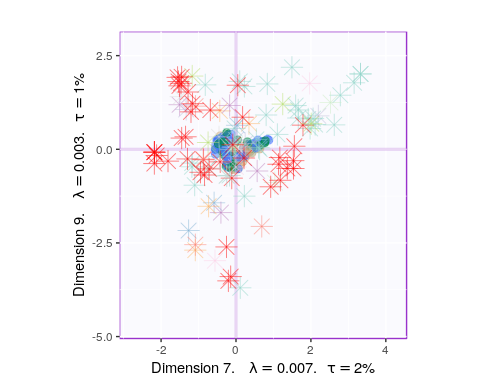
#### Symmetric Plot

for (i in 1:2){  
 axis1 = loop[i,1]  
 axis2 = loop[i,2]  
   
 symMap <- createFactorMapIJ(Fi,Fj,  
 col.points.i = color4Authors,  
 col.labels.i = color4Authors,  
 col.points.j = col4J,  
 col.labels.j = col4J, pch.j = 8, cex.j = 5, axis1 = axis1, axis2 = axis2)  
   
   
 labels4CA <- PTCA4CATA::createxyLabels.gen(axis1,axis2,lambda = country\_env\_mca$ExPosition.Data$eigs, tau = country\_env\_mca$ExPosition.Data$t) #createxyLabels(resCA = country\_env\_mca )  
   
   
 map.IJ.sym <- symMap$baseMap + symMap$I\_points + #+ symMap$I\_labels+symMap$J\_labels  
 symMap$J\_points + labels4CA  
 print(map.IJ.sym)  
}



#### Asymmmetric Plot

for (i in 1:2){  
 asymMap <- createFactorMapIJ(Fi,Fj.a,  
 col.points.i = color4Authors,  
 col.labels.i = color4Authors,  
 col.points.j = col4J,  
 col.labels.j = col4J, pch.j = 8, cex.j = 5, )  
   
 labels4CA <- PTCA4CATA::createxyLabels.gen(axis1,axis2,lambda = country\_env\_mca$ExPosition.Data$eigs, tau = country\_env\_mca$ExPosition.Data$t) #createxyLabels(resCA = country\_env\_mca)  
   
 axis1 = loop[i,1]  
 axis2 = loop[i,2]  
   
 map.IJ.asym <- asymMap$baseMap + #asymMap$I\_labels +   
 asymMap$I\_points + #asymMap$J\_labels +   
 asymMap$J\_points + labels4CA  
 print(map.IJ.asym)  
}



#### Most Contributing Variables (Inference)

Let’s plot variable contributions against each chosen components i.e. 1, 2, 7, 9.

* With Bootstrap Ratio

BR <- country\_env\_mca\_inf$Inference.Data$fj.boots$tests$boot.ratios  
  
for (i in c(1, 2, 7, 9)) {  
 laDim = i  
 ba001.BR1 <- PrettyBarPlot2(BR[,laDim],  
 threshold = 2,  
 font.size = 5,  
 color4bar = gplots::col2hex(col4J), # we need hex code  
 main = paste0('Bootstrap ratio ',laDim),  
 ylab = 'Bootstrap ratios',horizontal = FALSE  
 #ylim = c(1.2\*min(BR[,laDim]), 1.2\*max(BR[,laDim]))  
 )  
 print(ba001.BR1)  
}

