Unsupervised Learning Assignment 1

This notebook contains all the code for the unsupervised learning assignment 1 implementation of Multidimensional Scaling on the UN_Statistics dataset.

Multidimensional scaling (MDS) is a unsupervised, multivariate data analysis technique that is used to visualize the similarity/dissimilarity between samples by plotting points in two dimensional plots.

MDS represents data in a lower-dimensional space, mapped from a higher dimentional space. In this notebook one and two dimentional representations are generated for a higher dimentional data set.

From an alorithmic perspective, MDS uses a dissimilarity matrix to represent the distances between pairs of objects. The input data to the MDS algorithm is this disimilarity matrix.

Enviroment Setup and Import data files

This notebook is set up to make the results atained as reproducable as posible.

```r
#setup work space, install packages and import libs
rm(list=ls())
suppressMessages(library(cluster))
suppressMessages(library(MASS))
suppressMessages(library(smacof))
suppressMessages(library(magrittr))
suppressMessages(library(dplyr))
suppressMessages(library(ggpubr))
suppressMessages(library(psych))

#import and sample data, apply seed and sample to get the set of 400 unique data points
statistics <- read.csv("UN_Statistics.csv")
X <- as.matrix(statistics[,-1])
rownames(X) <- statistics[,1]
```

#data exploration

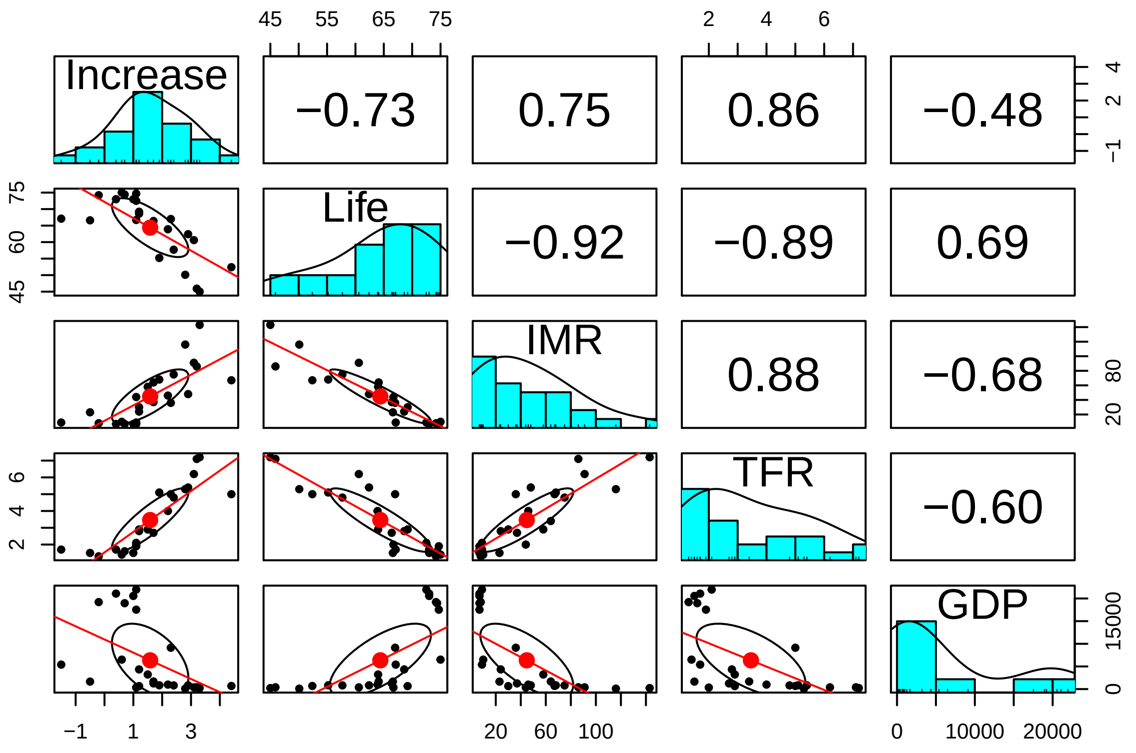
```r
rownames(X)
```

```
##  [1] "Albania"          "Argentina"        "Australia"
##  [4] "Austria"          "Benin"            "Boliva"
##  [7] "Brazil"           "Cambodia"         "China"
## [10] "Colombia"         "Croatia"          "El Salvador"
## [13] "France"           "Greece"           "Guatemala"
## [16] "Iran"             "Italy"            "Malawi"
## [19] "Netherlands"      "Pakistan"         "Papua new Guinea"
## [22] "Peru"             "Romania"          "USA"
## [25] "Zimbabwe"
```

```r
sapply(statistics,class)
```

```
##   Country  Increase      Life       IMR       TFR       GDP
##  "factor" "numeric" "numeric" "integer" "numeric" "numeric"
```

```
pairs.panels(statistics[,-1],cex=1,lm=TRUE)
```

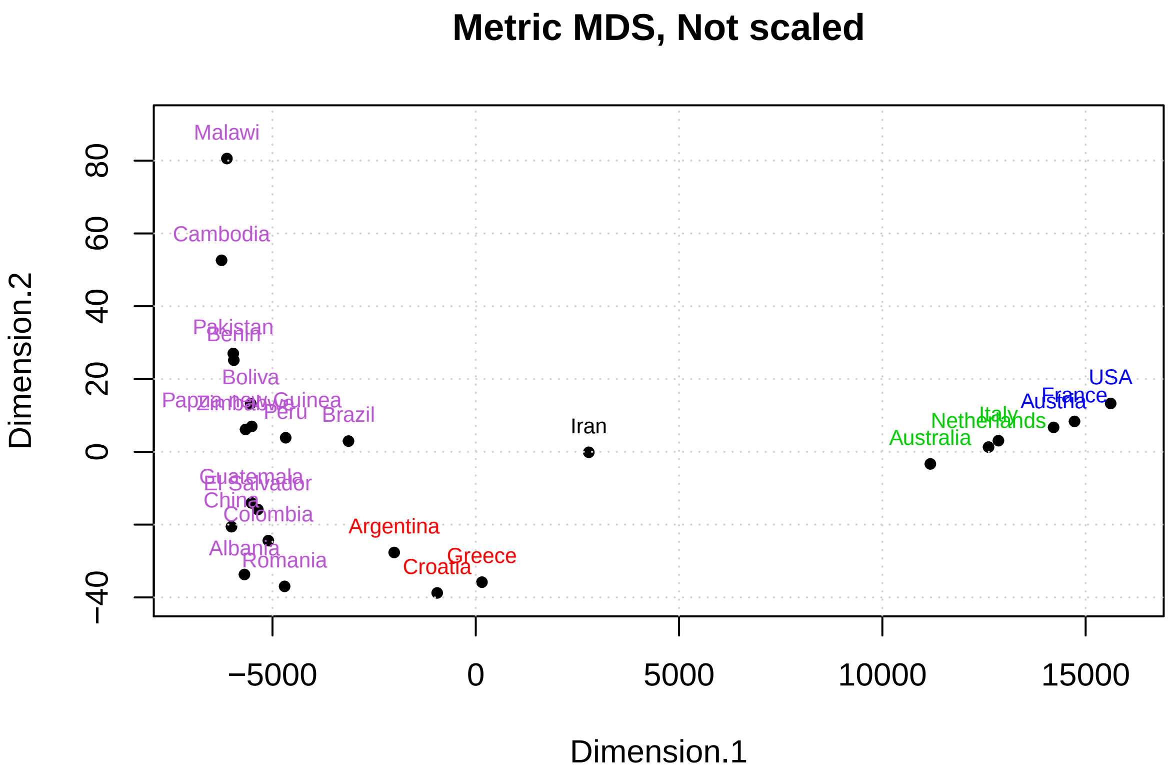


Classical MDS Scaling

First let's plot the data before we normalize it.

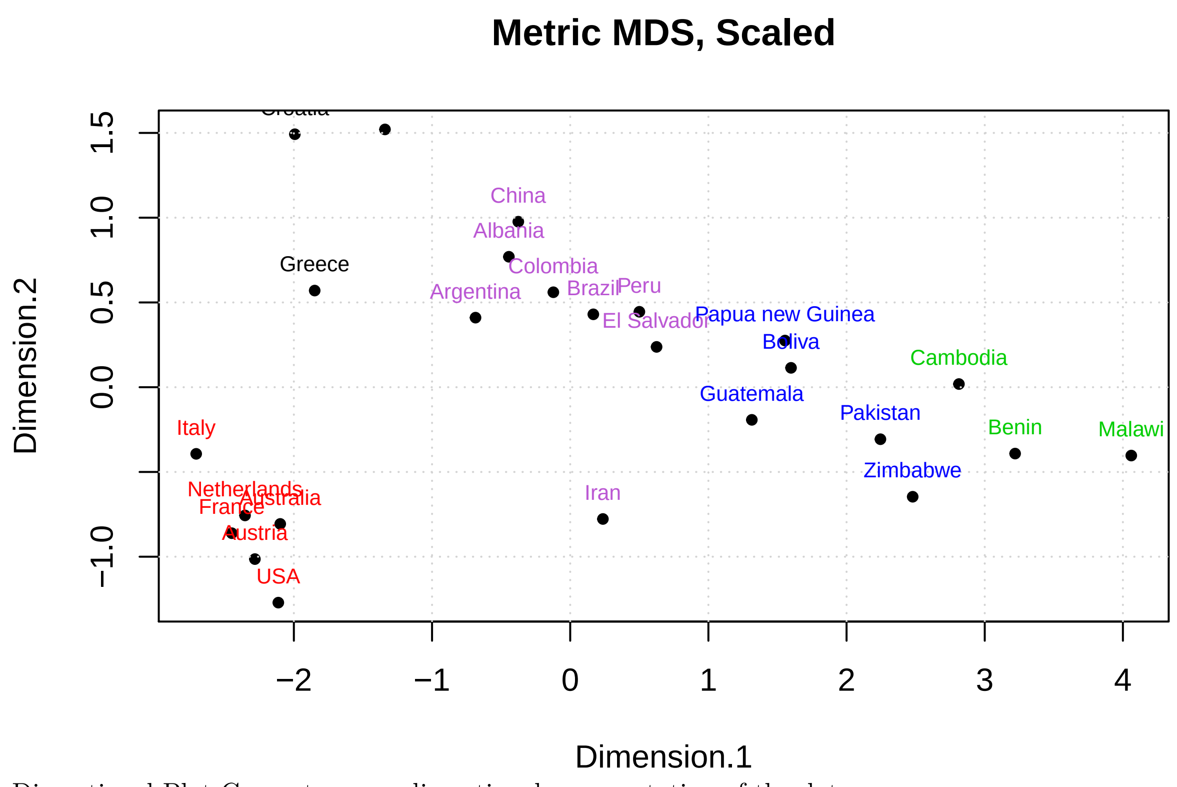
```
d <- dist(X) # euclidean distances between the rows

#preform clustering so we can allocate colours to the plot
cluster <- hclust(d,method="complete")
clusvec <- cutree(cluster, k=5)

scaledDistances <- cmdscale(d) # preform the multidimensional scalling

# create empty plot and then add text and colours. Colours added based on the clustor groups
plot (scaledDistances, xlab="Dimension.1", ylab="Dimension.2",
  main="Metric MDS, Not scaled", type="p", pch=20,ylim = c(-40,90), xlim = c(-7000,16000))
grid()
#ensure you list enough colours for the number of clusters
colvec <- c("mediumorchid",
            "red",
            "green3",
            "blue",
            "black",
            "gold",
            "indianred",
            "moccasin",
            "lightcyan",
            "skyblue")
```

```r
for (i in 1:length(scaledDistances[,1]))
  text (scaledDistances[i,1],
        scaledDistances[i,2],
        rownames(X)[i],
        col=colvec[clusvec[i]],
        cex=0.7,
        pos = 3)
```



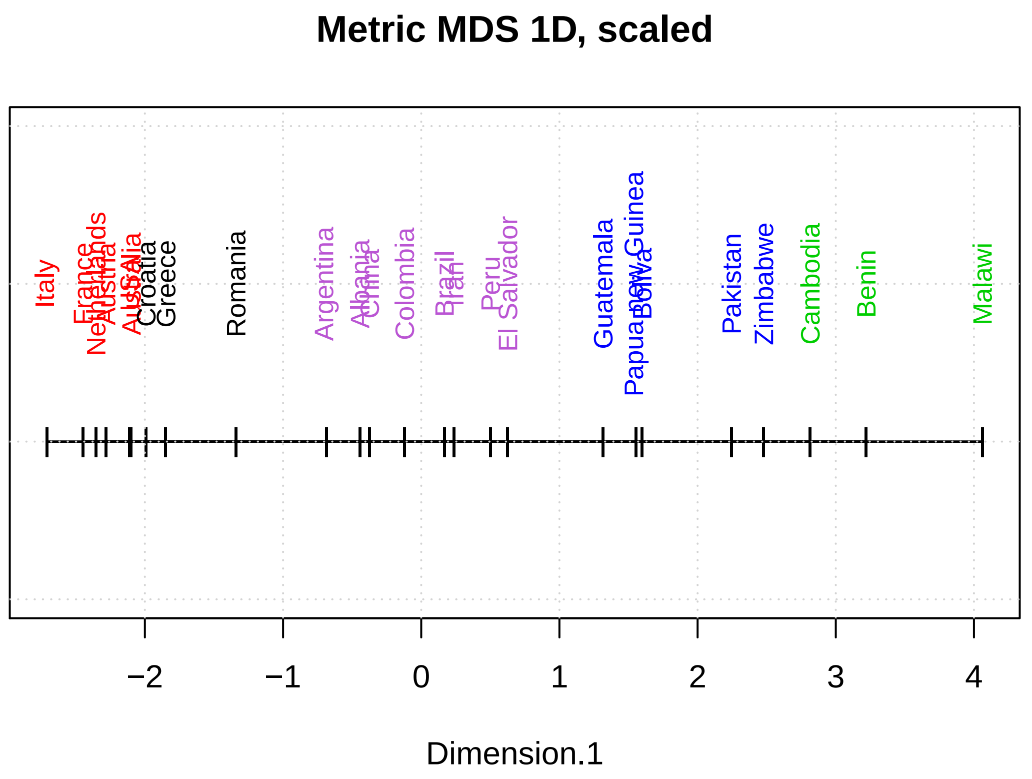
#Normalized MDS Next we can repeat the process but this time normalize the data scaling

```r
# euclidean distances between the rows
d <- dist(scale(X, center=TRUE, scale=TRUE), method="euclidean")

#preform clustering so we can allocate colours to the plot
cluster <- hclust(d,method="complete")
clusvec <- cutree(cluster, k=5)

scaledDistances <- cmdscale(d, k = 2) # preform the multidimensional scalling

# create empty plot and then add text and colours. Colours added based on the clustor groups
plot (scaledDistances, xlab="Dimension.1", ylab="Dimension.2",
  main="Metric MDS, Scaled", type="p", pch=20)
grid()
for (i in 1:length(scaledDistances[,1]))
  text (scaledDistances[i,1],
        scaledDistances[i,2],
        rownames(X)[i],
        col=colvec[clusvec[i]],
        cex=0.7,
        pos = 3)
```



#One Dimentional Plot Generate a one dimentional representation of the data

```
d <- dist(scale(X)) # euclidean distances between the rows

#preform clustering so we can allocate colours to the plot
cluster <- hclust(d,method="complete")
clusvec <- cutree(cluster, k=5)

scaledDistances <- cmdscale(d, eig=TRUE, k=1) # preform the multidimensional scalling

x <- data.frame(scaledDistances$points[,1],1)

# create empty plot and then add text and colours. Colours added based on the clustor groups
plot (x, xlab="Dimension.1",
  main="Metric MDS 1D, scaled",
  type = 'o',
  pch = '|',
  ylab = '',
  yaxt='n',
  xlim=c(min(x),
         max(x)),
  ylim=c(0.95,1.1))
grid()

for (i in 1:length(x[,1]))
  text (x[i,1],
        1.05,
        rownames(X)[i],
        col=colvec[clusvec[i]],
        cex=0.8,
        srt=90)
```

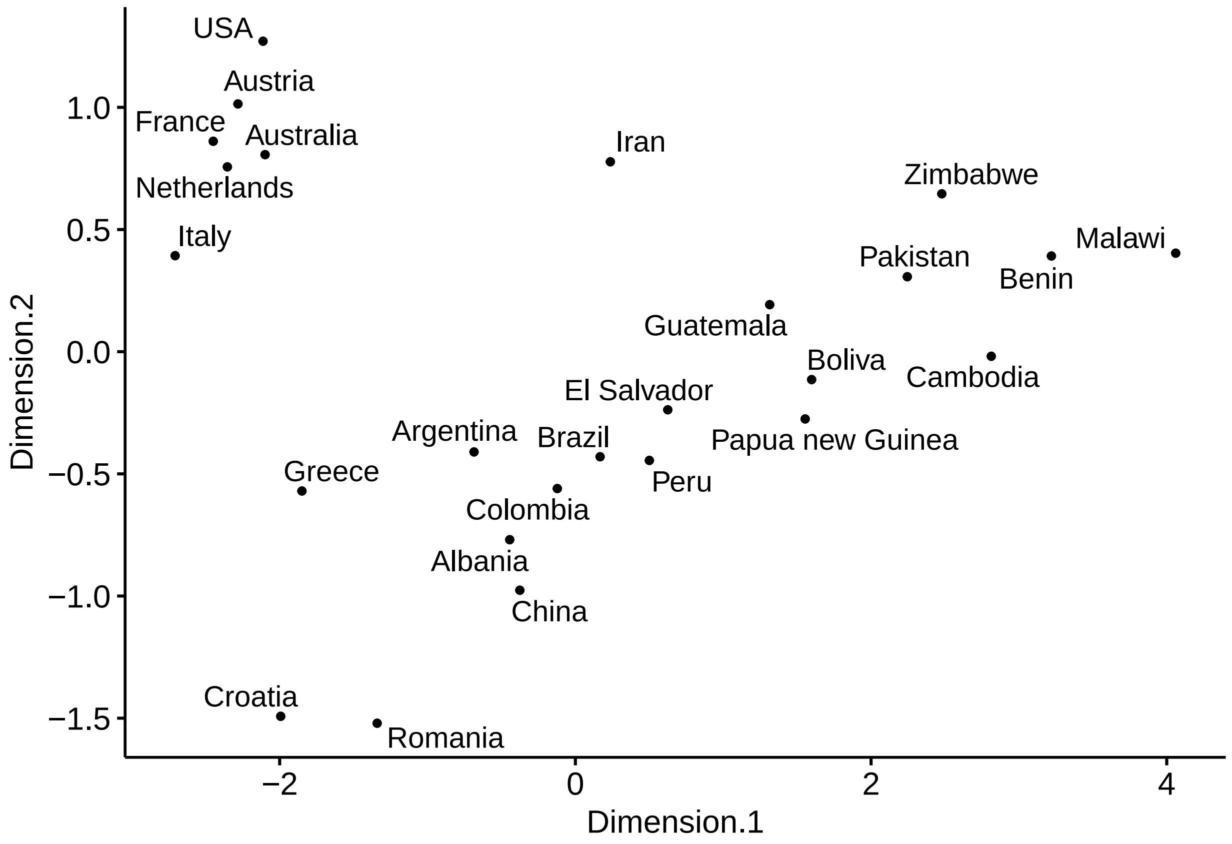


# Classical MDS

```
# Cmpute MDS
mds <- scale(X) %>%
  dist() %>%
  cmdscale() %>%
  as_tibble()
```

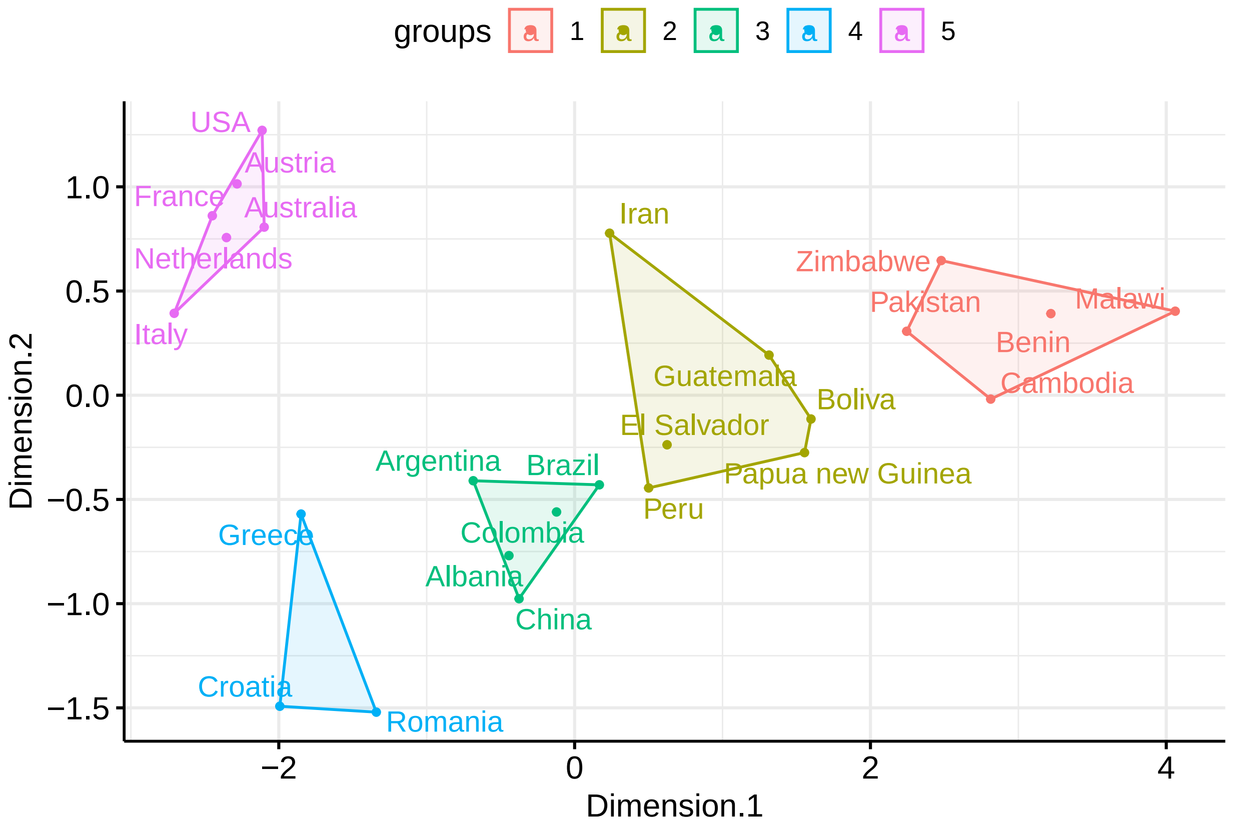
```
## Warning: `as_tibble.matrix()` requires a matrix with column names or a `.name_repair` argument. Usin
## This warning is displayed once per session.
```

```
colnames(mds) <- c("Dimension.1", "Dimension.2")
# Plot MDS
mds[,2] <- -1 * mds[,2]
ggscatter(mds, x = "Dimension.1", y = "Dimension.2",
          label = rownames(X),
          size = 1,
          repel = TRUE)
```



We can add colours to the plots while clustering them together

```
# K-means clustering
clust <- kmeans(mds, 5)$cluster %>%
  as.factor()
mds <- mds %>%
  mutate(groups = clust)
# Plot and color by groups
ggscatter(mds, x = "Dimension.1", y = "Dimension.2",
          label = rownames(X),
          color = "groups",
          palette = "pal3",
          size = 1,
          ellipse = TRUE,
          ellipse.type = "convex",
          repel = TRUE) +
grids(axis = c("xy", "x", "y"), color = "grey92", size = NULL,
  linetype = NULL)
```



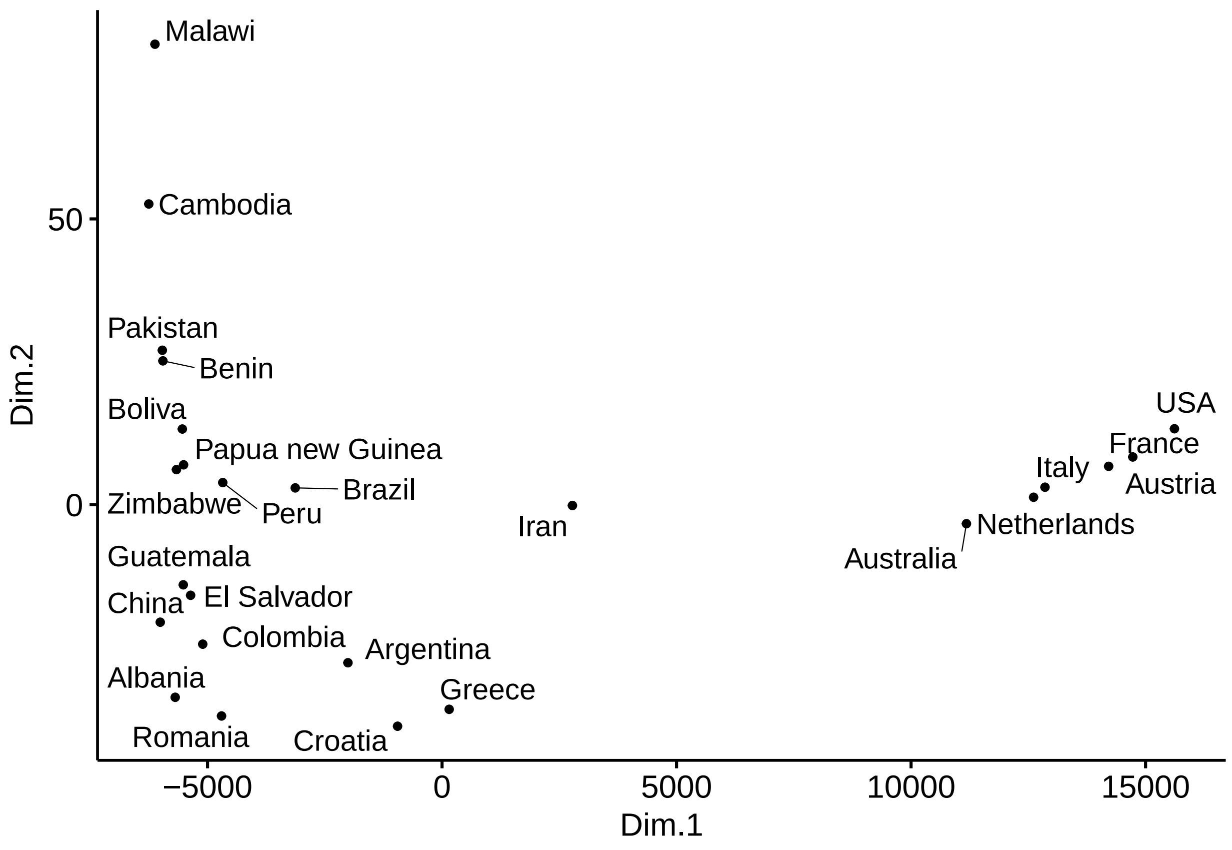
Non-metric MDS

Kruskal's non-metric multidimensional scaling

```
mds <- X %>%
  dist() %>%
  isoMDS() %>%
  .$points %>%
  as_tibble()
```

```
## initial  value 0.000168
## final  value 0.000168
## converged
```

```
colnames(mds) <- c("Dim.1", "Dim.2")
# Plot MDS
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(X),
          size = 1,
          repel = TRUE)
```

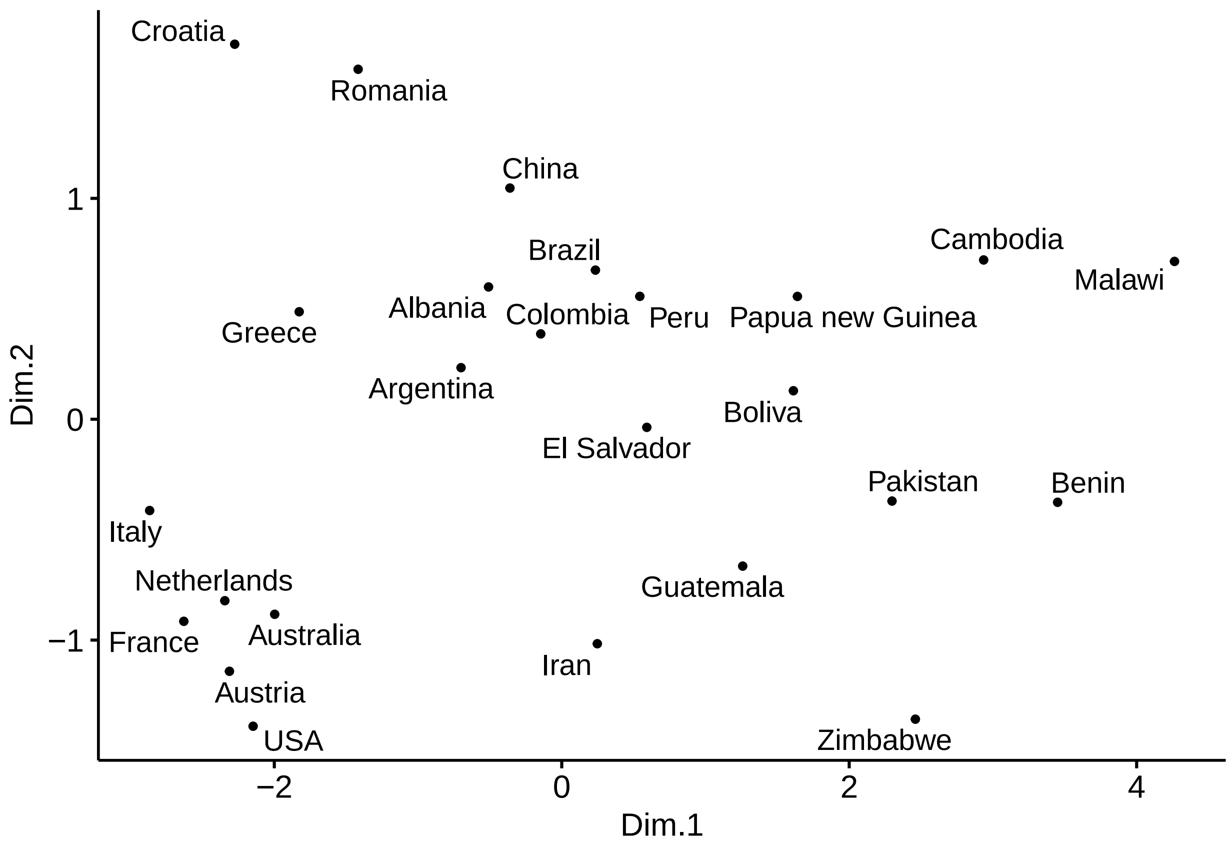


Sammon's non-linear mapping:

```r
mds <- scale(X) %>%
  dist() %>%
  sammon() %>%
  .$points %>%
  as_tibble()
```

```
## Initial stress        : 0.01485
## stress after  10 iters: 0.00425, magic = 0.500
## stress after  20 iters: 0.00421, magic = 0.500
```

```r
colnames(mds) <- c("Dim.1", "Dim.2")
# Plot MDS
ggscatter(mds, x = "Dim.1", y = "Dim.2",
          label = rownames(X),
          size = 1,
          repel = TRUE)
```

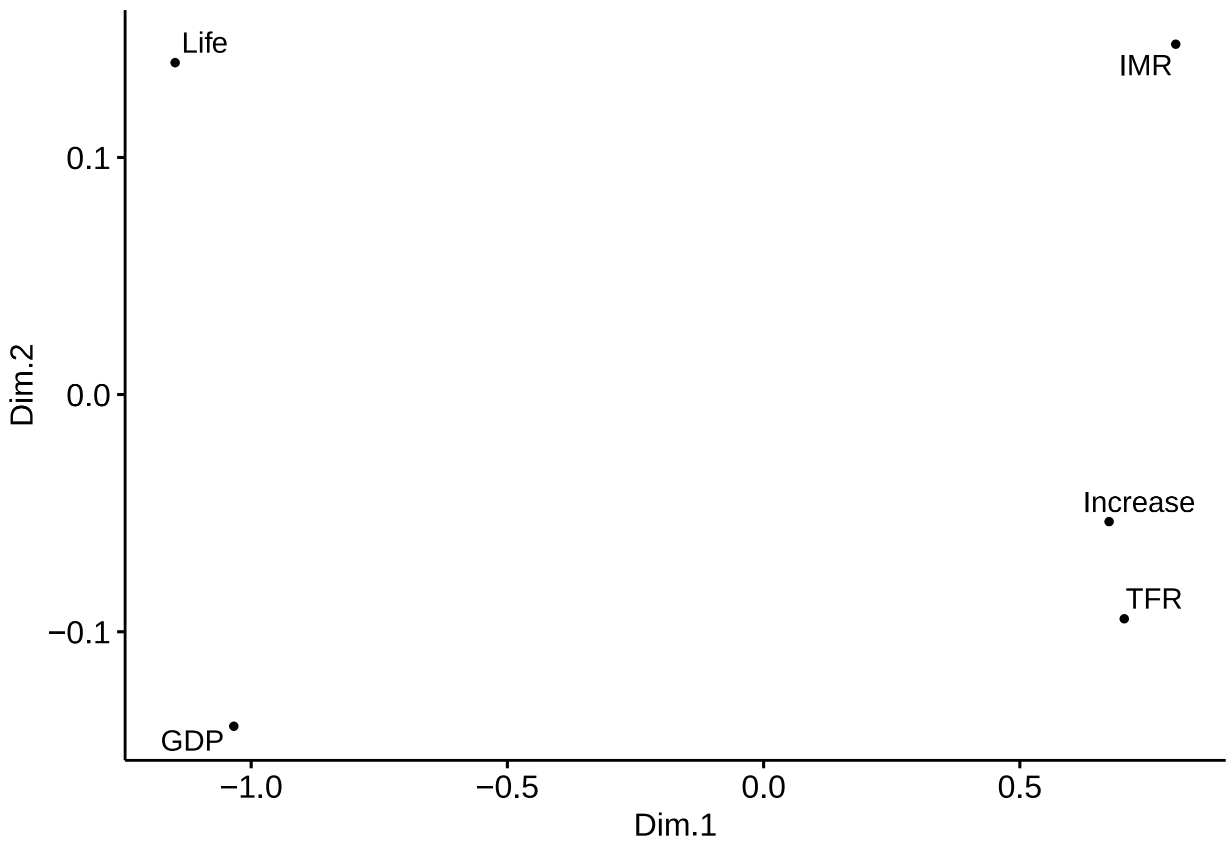


Visualizing a correlation matrix using Multidimensional Scaling

MDS can be also used to reveal a hidden pattern in a correlation matrix.

Correlation actually measures similarity, but it is easy to transform it to a measure of dissimilarity.

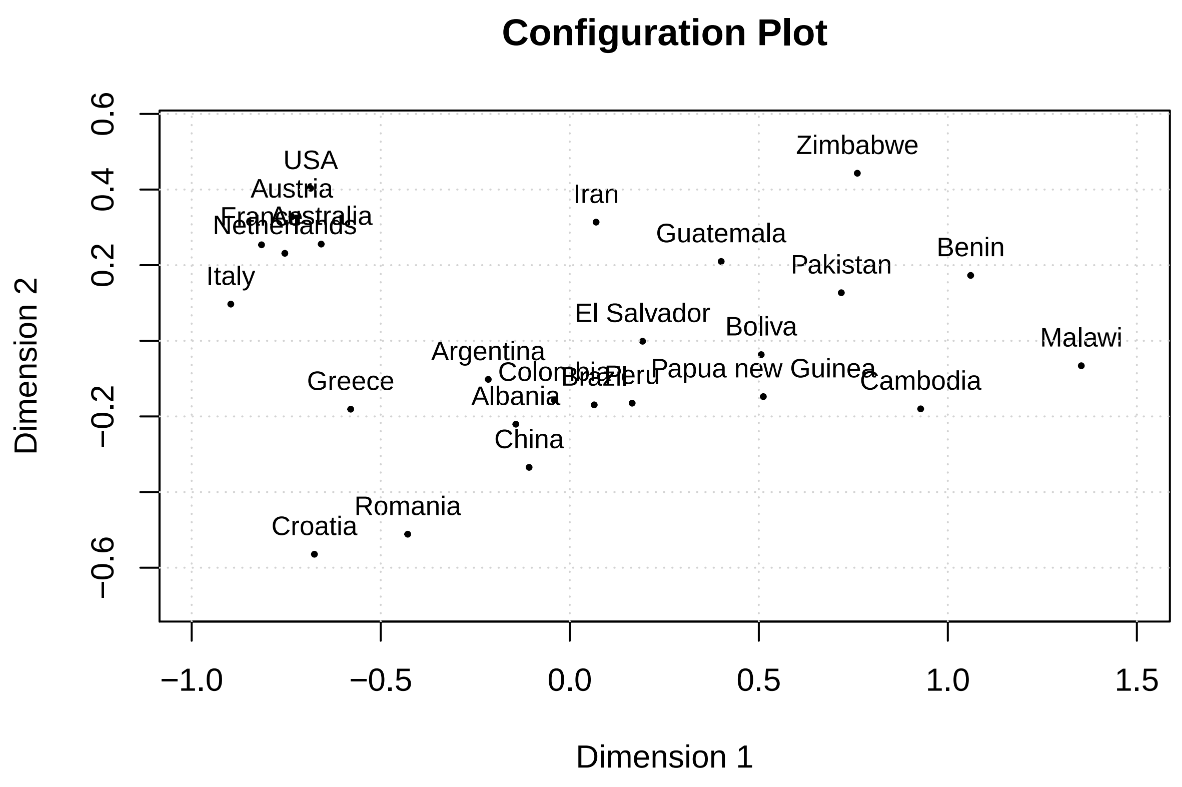
```
res.cor <- cor(X, method = "spearman")
mds.cor <- (1 - res.cor) %>%
  cmdscale() %>%
  as_tibble()
colnames(mds.cor) <- c("Dim.1", "Dim.2")
ggscatter(mds.cor, x = "Dim.1", y = "Dim.2",
          size = 1,
          label = colnames(res.cor),
          repel = TRUE)
```



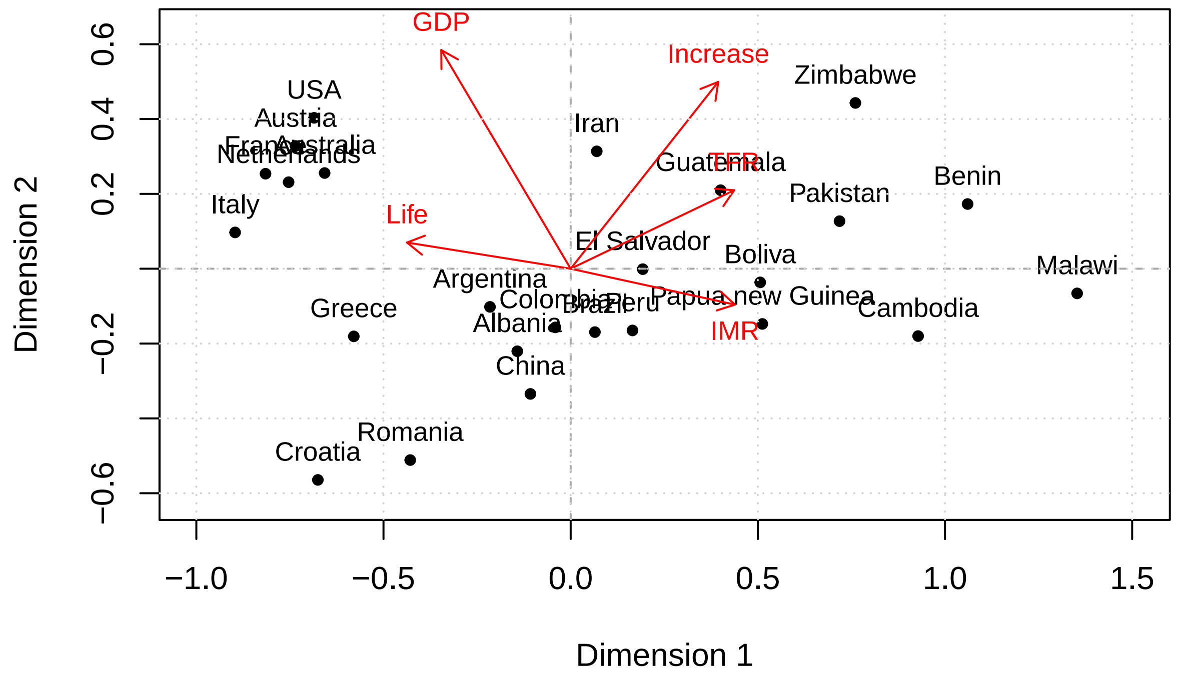
MDS Biplot

Visualize underlying variables along with MDS plot

```r
d <- dist(scale(X, center=TRUE, scale=TRUE), method="euclidean")

countriesMds <- mds(delta = d, ndim = 2)
plot(countriesMds)
grid()
```



```
biplot <- biplotmds(countriesMds, extvar = X)
plot(biplot,  vecscale = 0.3, vec.conf = list(col = "red", length = 0.1),main="")
grid()
```


GDP
Increase
TFR
Life
IMR
Dimension 1
Dimension 2
USA
Austria
France
Australia
Netherlands
Italy
Greece
Croatia
Romania
Argentina
Colombia
Brazil
Peru
Albania
China
Iran
Guatemala
El Salvador
Boliva
Papua new Guinea
Pakistan
Zimbabwe
Benin
Cambodia
Malawi