PRO1:

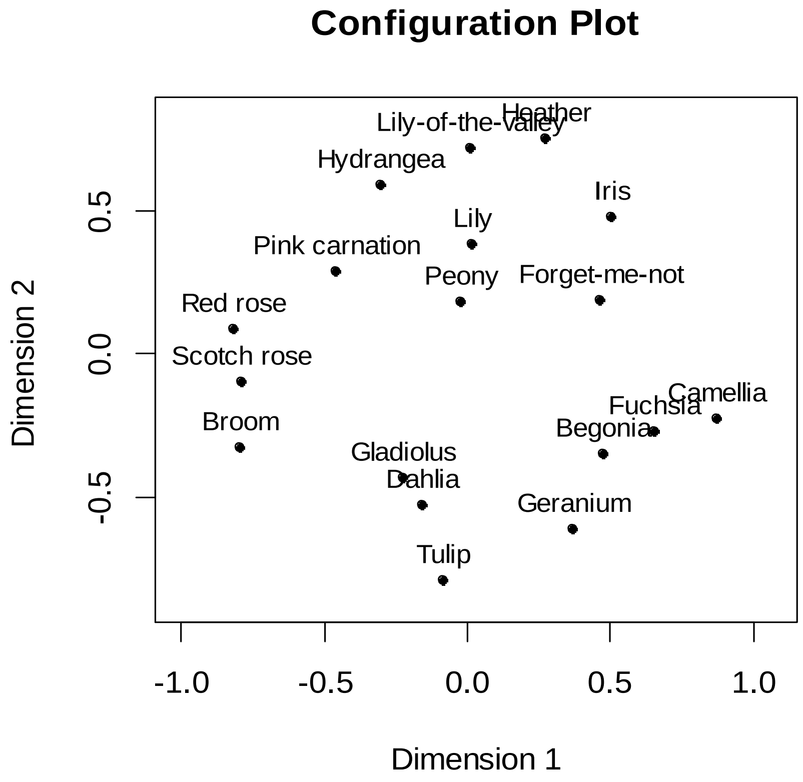
```
###HW4### code:


#problem1#


corANA=function(x){
    c=dim(x)[2];
    r=dim(x)[1];
    cmatrix=matrix(rep(0),c,c);
    rmatrix=matrix(rep(0),r,r);
    csum=apply(x,2,sum);
    rsum=apply(x,1,sum);
    total=sum(csum);
    #Chi-squared dist blt columns#
    for(i in 1:c){
      for(j in 1:c){
        for(k in 1:r){
          cmatrix[i,j]=cmatrix[i,j]+(total/rsum[k])*(x[k,i]/csum[i]-x[k,j]/csum[j])^2;
                }
              }
            }
    print("Chi-squared dist. matrix for columns:")
    print(sqrt(cmatrix));
    #Chi-square dist blt rows#
    for(i in 1:r){
      for(j in 1:r){
        for(k in 1:c){
          rmatrix[i,j]=rmatrix[i,j]+(total/csum[k])*(x[i,k]/rsum[i]-x[j,k]/rsum[j])^2;
                }
              }
            }
    print("Chi-squared dist.matrix for rows:")
    print(sqrt(rmatrix));
    }
```

PRO 2:

```
"flower"=structure(c(0.91, 0.49, 0.47, 0.43, 0.23, 0.31, 0.49, 0.57, 0.76,
0.32, 0.51, 0.59, 0.37, 0.74, 0.84, 0.94, 0.44, 0.67, 0.59, 0.9,
0.79, 0.7, 0.57, 0.57, 0.58, 0.77, 0.69, 0.75, 0.68, 0.54, 0.41,
0.2, 0.5, 0.59, 0.57, 0.29, 0.54, 0.71, 0.57, 0.58, 0.63, 0.69,
0.75, 0.68, 0.7, 0.75, 0.7, 0.79, 0.61, 0.52, 0.44, 0.26, 0.89,
0.62, 0.75, 0.53, 0.77, 0.38, 0.58, 0.37, 0.48, 0.48, 0.44, 0.54,
0.49, 0.5, 0.39, 0.46, 0.51, 0.35, 0.52, 0.54, 0.82, 0.77, 0.59,
0.24, 0.68, 0.61, 0.61, 0.52, 0.65, 0.63, 0.48, 0.74, 0.71, 0.83,
0.68, 0.49, 0.7, 0.86, 0.6, 0.77, 0.72, 0.63, 0.5, 0.61, 0.74,
0.47, 0.77, 0.7, 0.63, 0.47, 0.65, 0.49, 0.49, 0.64, 0.45, 0.22,
0.55, 0.46, 0.51, 0.35, 0.52, 0.36, 0.81, 0.77, 0.59, 0.47, 0.39,
0.41, 0.39, 0.52, 0.43, 0.38, 0.92, 0.36, 0.45, 0.37, 0.6, 0.84,
0.8, 0.59, 0.24, 0.17, 0.48, 0.62, 0.58, 0.67, 0.39, 0.39, 0.67,
0.62, 0.72, 0.49, 0.47, 0.57, 0.67, 0.45, 0.4, 0.61, 0.21, 0.85,
0.67), Size = 18, Diag = TRUE, Upper = FALSE, class = "dist", .Names = c("Begonia",
"Broom", "Camellia", "Dahlia",
"Forget-me-not", "Fuchsia", "Geranium",
"Gladiolus", "Heather", "Hydrangae",
"Iris", "Lily", "Lily-of-the-valley",
"Peony", "Pink carnation", "Red rose",
"Scotch rose", "Tulip",
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,
NA, NA, NA, NA, NA, NA, NA), Labels = c("Begonia",
"Broom", "Camellia", "Dahlia",
"Forget-me-not", "Fuchsia", "Geranium",
"Gladiolus", "Heather", "Hydrangea",
"Iris", "Lily", "Lily-of-the-valley",
"Peony", "Pink carnation", "Red rose",
"Scotch rose", "Tulip"))
library(smacof)
ns.fit=mds(data,type="ordinal")
plot(ns.fit)
```



We can find from the scatterplot:

GROUP 1: Lily-of-vally, heather, Hydrangea, Iris, Lily, Pink carnation, Peony, Forget-me-not might share some common characteristics.

GROUP 2: Red rose, Scotch rose, Broom might share some common characteristics.

GROUP 3: The other kinds of flowers (Tulip, Gladiolus etc.) might share some common characteristics.

Pro 3:

Code:

```
##problem 3###

library(HSAUR2)

data(USairpollution)

data=USairpollution

try.mds=cmdscale(dist(data),k=dim(data)-1,eig=T)

try.mds$eig
```

```
#The result is not so well. Reason: euclidean distance is not a good choice#

#why don't we use Mahalanobis distance? It's a better choice!#

#Covariance Matrix#

S=var(data)

md=apply(data,1,function(i) mahalanobis(data,i,S))

try2.mds=cmdscale(md,k=dim(data)[1]-1,eig=T)

cumsum(abs(try2.mds$eig))/sum(abs(try2.mds$eig))

#The result is not well, either. Too many dimensions are needed!#

#Try some non-metric scaling?

try3.nm=mds(md,type="ordinal")

#Guess it won't give us a good recovery of the map of these cities

#But the relative far or near location among them should be well-demonstrated!

#Cause Mahalanobis distance give us a good sense of dissimilarities

library(ggmap)

map=get_map(location='America',zoom=5)

plot(map)

par(new=T)

plot(try3.nm,col="red")
```

First try: Using Euclidean distance to generate similarity matrix and using classical multidimensional scale.

The result shows we need only on dimensions to have a good recovery of the initial configuration of the data.

Analysis: The Euclidean distance might not be a good choice because such a similarity matrix cannot contain enough information of the data (i.e. the index which relative large scale, like “manu”, have large contribution to the similarity (or we can say dissimilarity) among different cities). The reason why we only get the result that one dimension is enough is the equivalence between the classical multidimensional scale and principle component analysis. If we use the Euclidean distance, the result is the same as that derived from the PCA on covariance matrix. So, it won’t be a good choice. We consider to use Mahalanobis distance, which might have a offset on the influence caused by the different scales among indexes.

Second try: Using Mahalanobis distance and performing CMDS.

(R: result)

```
> cumsum(abs(try2.mds$eig))/sum(abs(try2.mds$eig))
 [1] 0.1553339 0.2888651 0.4069348 0.4906460 0.5725306 0.6498147 0.7154796
 [8] 0.7154796 0.7154817 0.7154927 0.7155079 0.7155506 0.7156059 0.7156927
```

The result shows we need more than three dimensions to have a good recovery on the initial configuration. However, we would like to use two dimensions so that we can visualize it.

Analysis: What if we use non-metric MDS on the dissimilarities distance generated by Mahalanobis distance? That is to say, we treat these dissimilarities as the original guess (or fitted distance) between different cities. If so, we can use the order(rank) of these dissimilarities and isotonic regression to generated a more reasonable or smoothing distance matrix, and recover the data based on the generated distance matrix.

Third try: Using Mahalanobis distance matrix and perform non-metric MDS.

Result: the stress index here is about 18%, which means it won't be a bad idea to do so (though the stress here is relatively high).

Initial guess:

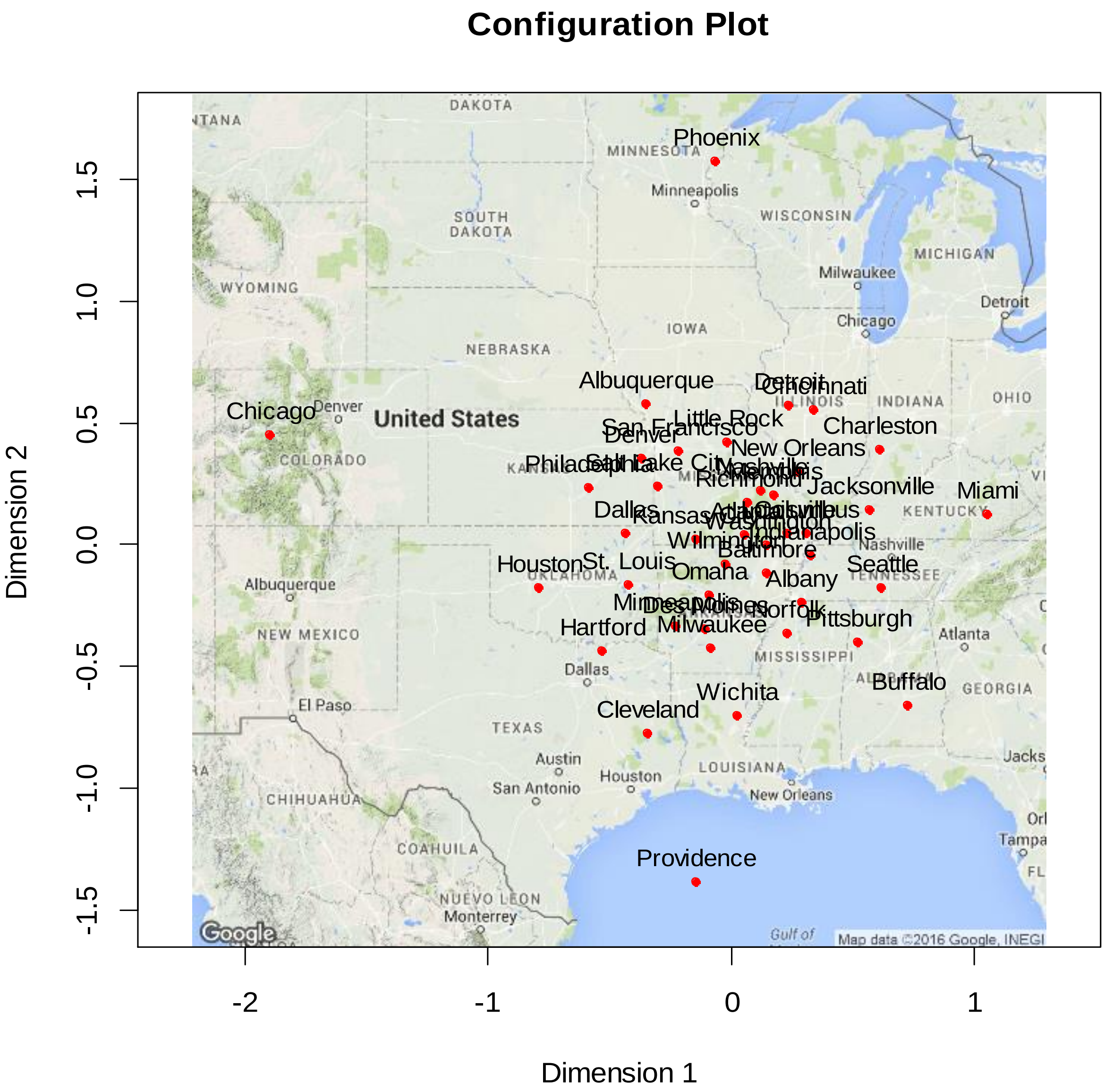
The recovery based on the non-metric MDS will not have a good recovery on the relative locations of these cities on the map, because the dissimilarities matrix is not calculated by using the relative distance between these cities. However, as the relative distance between these cities will have some influence on the air pollution conditions on these cities. So this two-dimensional recovery will reflect some information about the relative distance among these cities, but won't be very precious.

The result will be like this:

1. The groups, whose air pollution conditions are similar in some sense, will be distinctly shown in the recovered configuration.

2. The relative distance among these cities will be somewhat recovered, but only the distance, not the relative direction.

3. The distance between different groups will be presented in the configuration, but won't be so precious as that in map.



As we can see some concentric-circle-like patterns in the plot (i.e. the Phoenix, Miami, Chicago, Providence Buffalo, lay on the outer circle of these cities, because the show much difference compared to other cities, for example, Miami is high in "temp" index, while others, except the other four stated above, do not show much high on "temp" index). The distinct groups are characterized by different concentric-circle.

And the relative distance is not shown very well. Like San Francisco is more distant from Miami than Chicago, but we cannot find any information in the plot shows their relative distance (In fact, the plot shows San Francisco has the same distance to Miami and Chicago, which is not reasonable, after all).