W4415 Multivariate Statistical Inference Homework 4

Q1

```r
#calculate the chi-squared distance matrices for
#both rows and columns in a two-dimensional contingency table
chisqds<-function (x)
{
  # stop the function if x is not a matrix
  stopifnot(is.matrix(x))
  n<-nrow(x)
  p<-ncol(x)
  nsum<-rowSums(x)
  psum<-colSums(x)
  total<-sum(nsum)
  # preallocate empty vector
  ncij<-matrix(0,nrow=n,ncol=p)
  pcij<-matrix(0,nrow=n,ncol=p)
  disn<-matrix(0,nrow=n,ncol=n)
  disp<-matrix(0,nrow=p,ncol=p)
  # calculate weights for rows
  for(i in 1:n)
  {
    pcij[i,]<-x[i,]/nsum[i]
  }
  # calculate weights for columns
  for(i in 1:p)
  {
    ncij[,i]<-x[,i]/psum[i]
  }
  # calculate the chi-squared distance matrices for rows
  for(i in 1:n)
  {
    for(j in 1:(i-1))
    {
      chisqd<-sqrt(sum(total/psum*(pcij[i,]-pcij[j,])^2))
      disn[i,j]<-chisqd
    }
  }
  # calculate the chi-squared distance matrices for columns
  for(i in 1:p)
  {
    for(j in 1:(i-1))
    {
      chisqd<-sqrt(sum(total/nsum*(ncij[,i]-ncij[,j])^2))
      disp[i,j]<-chisqd
    }
  }
  chisqds<-list(disn,disp)
}
```

```r
examp<-matrix(c(18,42,7,33),nrow=2,ncol = 2)
chis<-chisqds(examp)
# row distance
chis[1]
```

```
## [[1]]
##           [,1] [,2]
## [1,] 0.0000000    0
## [2,] 0.3265986    0
```

```r
# column distance
chis[2]
```

```
## [[1]]
##           [,1] [,2]
## [1,] 0.0000000    0
## [2,] 0.2886751    0
```

```r
library(ade4)
# compare with result of chi-square distance built-in function
# dudi.coa from package {ade4}, the row distance is the same
dist.dudi(dudi.coa(examp,scannf=FALSE,nf=2),amongrow=TRUE)
```

```
##           1
## 2 0.3265986
```

Q2

```r
library("HSAUR2")
library(smacof)
library(scatterplot3D)
data("gardenflowers")
fl_mds<-cmdscale(gardenflowers,k=17,eig=TRUE)
## Warning in cmdscale(gardenflowers, k = 17, eig =TRUE): only 9 of the first
## 17 eigenvalues are > 0

eigenfl<-fl_mds$eig
eigenfl[1:10]
```

```
##  [1]  1.173085e+00  8.944353e-01  5.732009e-01  4.843006e-01  2.638516e-01
##  [6]  2.298165e-01  8.383417e-02  6.645861e-02  2.984017e-02 -1.387779e-16
```

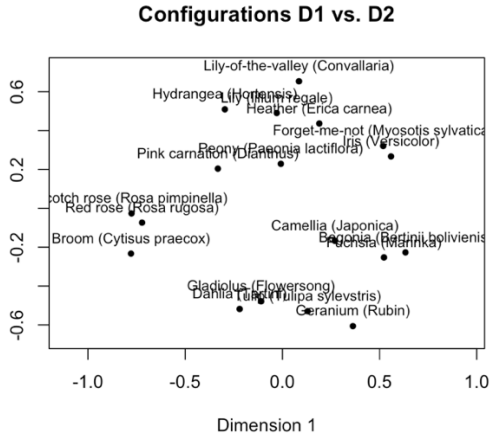
```r
# assess how many coordinates needed to adequately represent
# the observed distance matrix
# criterion Pm(1)
# use absolute value as some of the eigenvalues are negatives
cumsum(abs(eigenfl))/sum(abs(eigenfl))
```

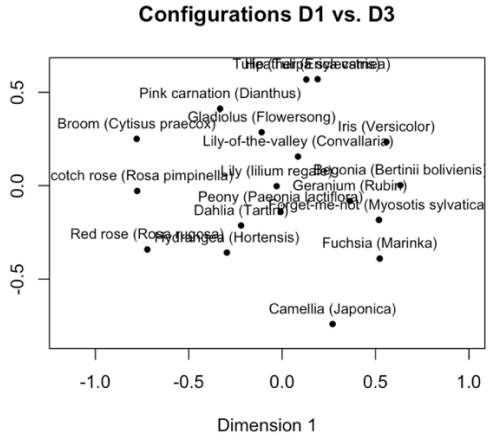
```
##  [1] 0.2549273 0.4493002 0.5738645 0.6791096 0.7364481 0.7863903 0.8046086
##  [8] 0.8190510 0.8255356 0.8255356 0.8302034 0.8391005 0.8485894 0.8678594
## [15] 0.8905760 0.9183028 0.9555567 1.0000000
```

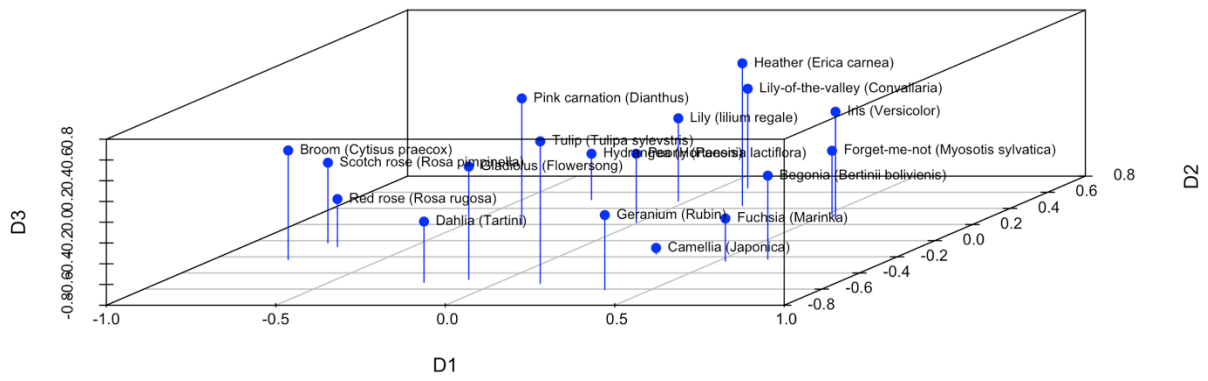
```
# criterion Pm(2)
cumsum(eigenfl^2)/sum(eigenfl^2)

##  [1] 0.4611160 0.7291863 0.8392805 0.9178730 0.9412006 0.9588982 0.9612532
##  [8] 0.9627331 0.9630315 0.9630315 0.9631861 0.9637478 0.9643866 0.9670214
## [15] 0.9706829 0.9761377 0.9859851 1.0000000

x<-fl_mds$points[,1]
y<-fl_mds$points[,2]
plot(x,y,xlab = "Coordinate 1", ylab = "Coordinate 2",
     xlim = range(x)*1.2, type = "p")
text(x, y, labels = attr(gardenflowers,'Labels'), cex = 0.7)
fit<-mds(gardenflowers,ndim=3)
par(mfrow=c(1,2))
plot(fit, plot.dim = c(1,2), main = "Configurations D1 vs. D2")
plot(fit, plot.dim = c(1,3), main = "Configurations D1 vs. D3")
s3d = scatterplot3d(fit$conf[,1],fit$conf[,2], fit$conf[,3],color="blue", pch
=19,angle = 70, type="h", xlab="D1",ylab="D2",zlab="D3")
s3d.coords = s3d$xyz.convert(fit$conf[,1], fit$conf[,2], fit$conf[,3])
text(s3d.coords$x, s3d.coords$y,labels=dimnames(fit$conf)[[1]], cex=.7, pos=
4)
fit$stress
[1] 0.1408848
```

At first, I use classical cmdscale() function to determine the appropriate dimensions. The two criteria for judging number of dimensions differ considerably, but both values suggest that the original distances between the flower populations can be represented adequately in 3 dimensions. So I use the smacof() function to do multidimensional scaling and plot the results.

From the result of stress as 0.14, the MDS could be interpreted as a fair scaling. Plotting 2D pictures with dimension 1,2, and 3, and it verifies that the 3 dimension is also important as data points vary their relative positions in 2 plots.







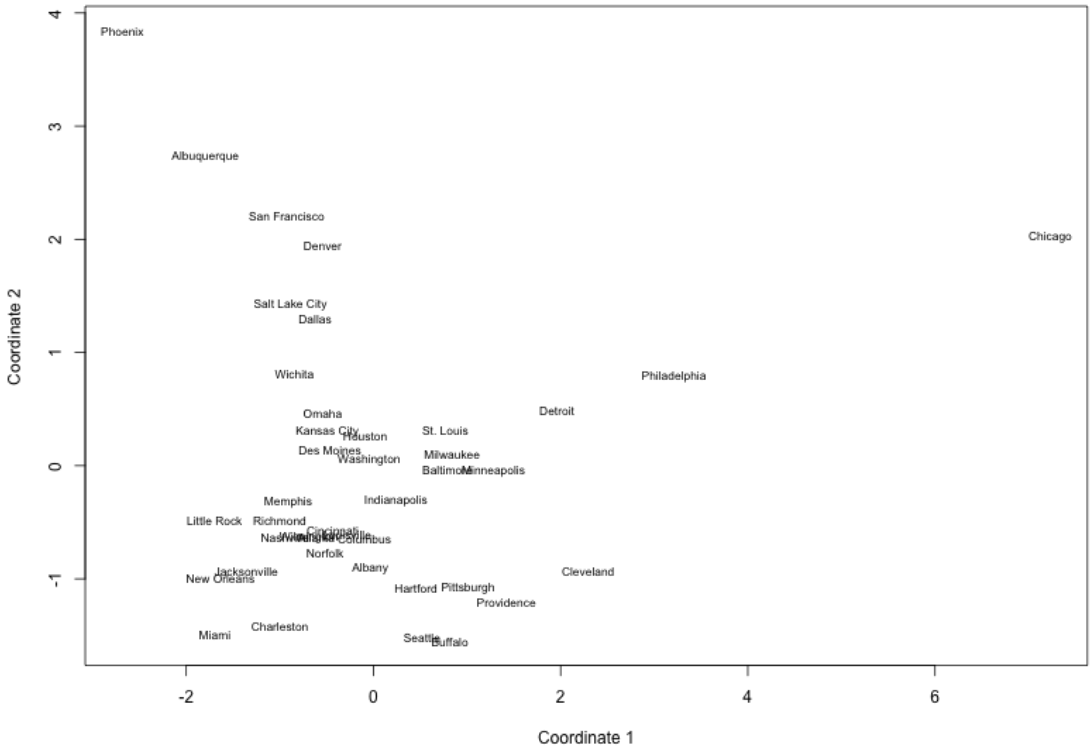
(1) From the plots above, it is easy to see that Broom, Scotch rose and Red rose points are close to each other, while the latter two roses are both Rosa. Though they have different shapes, they may share some common properties.
(2) Dahila and Gladiolus is close to each other, which have a distance of 0.24 in the original matrix.
(3) Begonia and Fuchsia have almost the same number in dimension 2, suggesting that they have properties in common.
(4) Forget-me-not, Iris also may share some common properties.

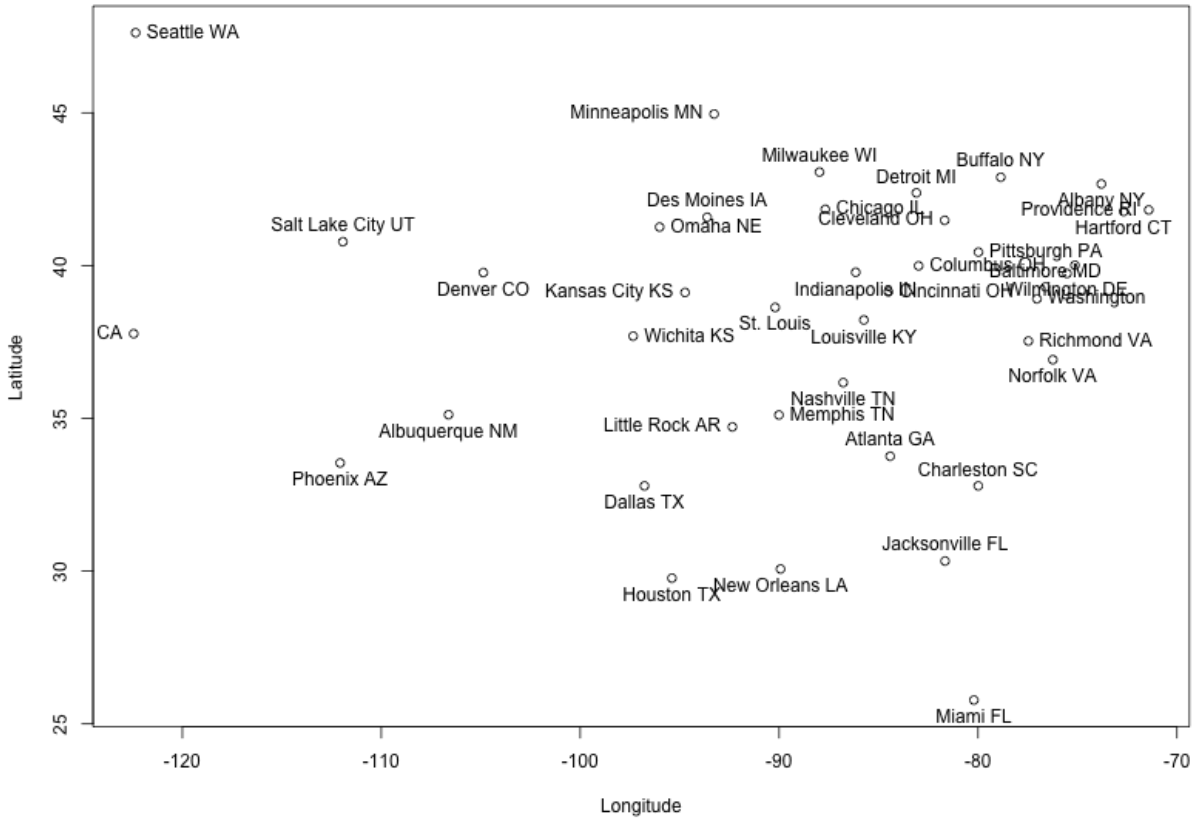
Q3

```
data("USairpollution")
library(maps)
library(RgoogleMaps)
library(ggmap)
apdiss<-dist(scale(USairpollution))
ap_mds<-cmdscale(apdiss,k=40,eig=TRUE)
## Warning in cmdscale(apdiss, k = 40, eig = TRUE): only 21 of the first 40
## eigenvalues are > 0

ap_mds$eig[1:10]

##  [1]  1.091248e+02  6.049339e+01  5.579892e+01  3.567965e+01  1.387115e+01
##  [6]  4.011504e+00  1.020597e+00  1.163285e-14  5.195944e-15  4.928423e-15

# criterion Pm(1)
cumsum(ap_mds$eig[1:10])/sum(ap_mds$eig)

##  [1] 0.3897314 0.6057792 0.8050611 0.9324884 0.9820282 0.9963550 1.0000000
##  [8] 1.0000000 1.0000000 1.0000000

# criterion Pm(2)
cumsum((ap_mds$eig[1:10])^2)/sum(ap_mds$eig^2)

##  [1] 0.5905751 0.7720615 0.9264731 0.9896080 0.9991503 0.9999483 1.0000000
##  [8] 1.0000000 1.0000000 1.0000000
```

```r
x<-ap_mds$points[,1]
y<-ap_mds$points[,2]
plot(x,y,xlab = "Coordinate 1", ylab = "Coordinate 2",
     xlim = range(x)*1.2, type = "n")
text(x, y, labels = rownames(USairpollution), cex = 0.7)

apfit<-mds(dist(scale(USairpollution)))
# find the location of cities
data("us.cities")
adr<-vector()
for(i in 1:41)
{
  adr<-rbind(adr,us.cities[which(grepl(
    rownames(USairpollution)[i],us.cities[,1])),c(1,4,5)])
}
adr<-adr[-c(1,3,6,8,11,12,15,17,18,23,25,27,31,33,35,38,40,41,51,53,54,58,59,
61,64),]
getGeoCode("St. Louis")

##      lat      lon
##  38.6270 -90.1994

getGeoCode("Washington")

##       lat       lon
##  38.90719 -77.03687

adr<-rbind(adr,c("St. Louis",38.63,-90.20))
adr<-rbind(adr,c("Washington",38.91,-77.04))
x<-adr[,3]
y<-adr[,2]
plot(x,y,xlab = "Longitude", ylab = "Latitude",type = "p")
text(x,y,labels = adr[,1], cex = 0.7)
```

Before MDS the US air pollution data, I center the matrix to ensure that scale of a certain variable may not significantly affect the distance matrix. The first 2 dimensions suggest a goodness of fit at 60% and 77% at 2 criteria, which is enough to retain information. The plot representing their relative distance on the first 2 dimensions is shown below.





Some surprising findings:

(1) Starting analysis from the outliers, Chicago is quite different from others. Though it is surrounded by Detroit, Cleveland and Milwaukee in geographical sense, it is far away from them in terms of air pollution indicators. The reason behind that might be its reliance on transportation industry, indicated by its large number of manufacturing enterprises (variable "manu"), which can explain its high level SO2 emission. For example, GE, Boeing and United Airlines have their headquarters in Chicago. Apart from that, it has the 3rd largest population in the US, which is 2,830,144 in the dataset(us.cities) belonging to package(map), while Detroit, has 871,789, approximately 2,000,000 smaller than that.
(2) Philadelphia, while near Wilmington, Washington and Baltimore in geographical map, shows different air pollution pattern. With its 5th-most-populus (quoted from Wiki) and 7th-largest metropolitan economy in the US, no wonder it has a higher level of SO2 compared with its neighbor cities.
(3) San Francisco and Denver, though far away from each other in geophysical map, display similar pattern in air pollution. Because Denver has an important role in agricultural industry, and San Francisco is famous for its high-tech industry, both don't rely on manufacturing enterprises much, leading to a low emission of SO2.