Multidimensional Scaling in Life: Creating distance matrices for multi cities in the world

Jingyi Cui

Due: 6/5

Contents

1 Introduction

While working with multidimensional scaling, the first data set I use the same data set as homework #4. The distances are close together (farthest apart no more than 5000 km apart). Ten second data set is six cities that are far apart: Beijing, Nairobi, Wellington, Brasilia, Seattle and Oslo. And most distances are far apart. I will use R to solve this problem. Below are the two mileages chart between these ten cities(in kilometers). The distances between two cities I collect below are using the air mileage.

cities / cities	Beijing	Dali	Shenzhen	Hong Kong	Ürümqi	Nanjing	Suzhou	Guilin	Chongqing	Wuhan
Beijing		2562	1943	1971	2413	897	1028	1723	1459	1054
Dali	2562		2421	2431	1604	2649	2825	1952	1493	2227
Shenzhen	1943	2421		17	3386	1155	1169	490	1082	894
Hong Kong	1971	2431	17		3402	1169	1180	504	1097	910
Ürümqi	2413	1604	3386	3402		3006	3196	2907	2305	2763
Nanjing	897	2649	1155	1169	3006		191	1119	1199	459
Suzhou	1028	2825	1169	1180	3196	191		1210	1359	608
Guilin	1723	1952	490	504	2907	1119	1210		602	707
Chongqing	1459	1493	1082	1097	2305	1199	1359	602		750
Wuhan	1054	2227	894	910	2763	459	608	707	750	

Table 1: Mileage Chart

cities / cities	Beijing	Wellington	Nairobi	Brasilia	Seattle	Oslo
Beijing		10773	9209	16927	8684	7020
Wellington	10773		13656	12287	11652	17660
Nairobi	9209	13656		9400	14478	7159
Brasilia	16927	12287	9400		10174	9907
Seattle	8684	11652	14478	10174		7324
Oslo	7020	17660	7159	9907	7324	

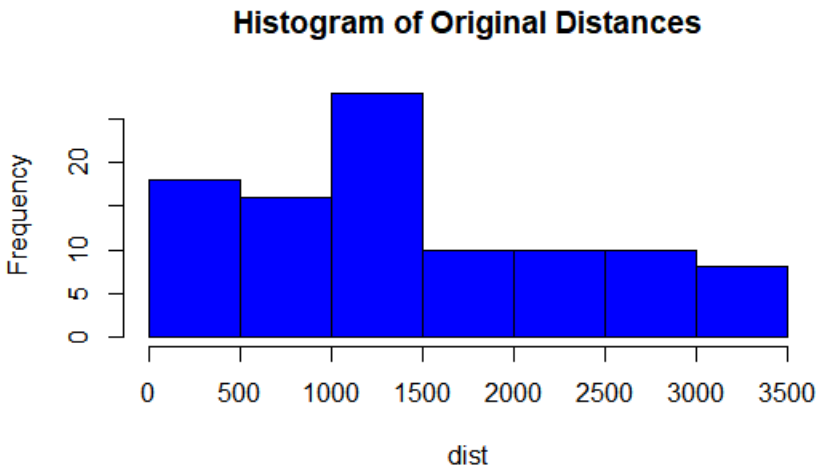
Table 2: Mileage Chart

These two distance matrices are D and E in which D_{ij} and E_{ij} are distances between object i and j. For the purpose of MDS, we use the symmetric distance functions, so $D_{ij} = D_{ji}$ and $D_{ii} = 0$, same with matrix E, so the blank space in each matrix is 0 while being the input to R. .

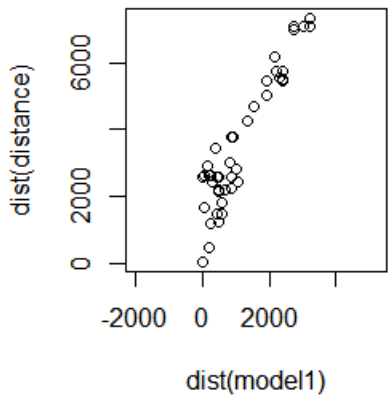
2 Results and Conclusion

2.1 Data Set 1

Here is the histogram of the original distances between the ten cities. We can observe that most distances between two cities are in range of 1000 to 1500.

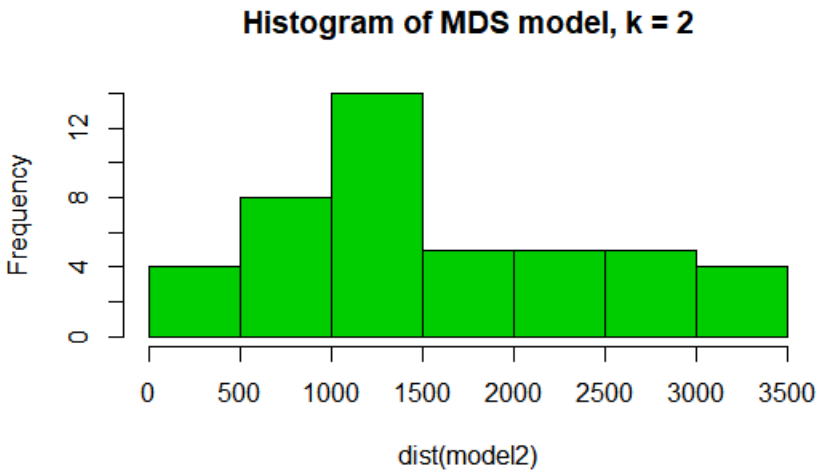


First, I try the 1 dimensional model. The output is a list of 1-dimensional coordinates. I plot the result and the original distances as known as "$y = x$" graph:

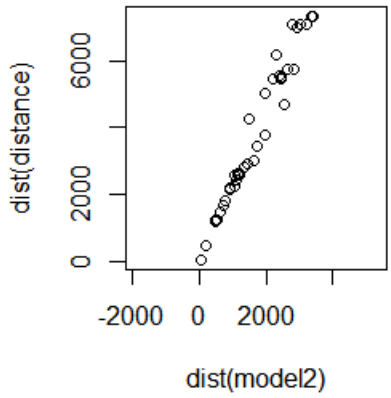


Then, to investigate how the good is the model, I get the GOF the R gives: 0.72. Since a value of 1 indicates a perfect fit, so it looks like that 0.72 is not bad. We will discover more in 2,3 dimensional models.

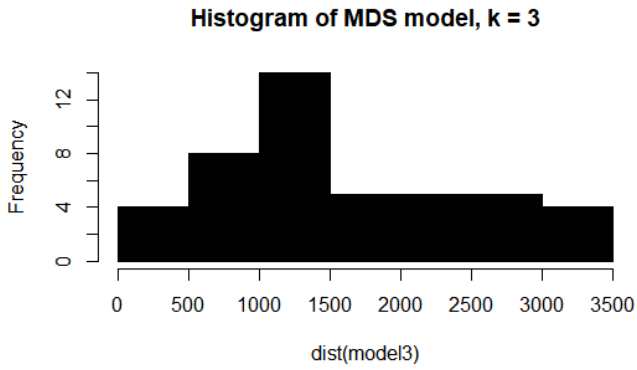
Next, I try 2 dimensional model. The output is a list of 2-dimensional coordinates. Here is the histogram of the distances calculated and we notice that its shape is similar to the histogram of original distances, with most frequency in range of 1000 to 1500:

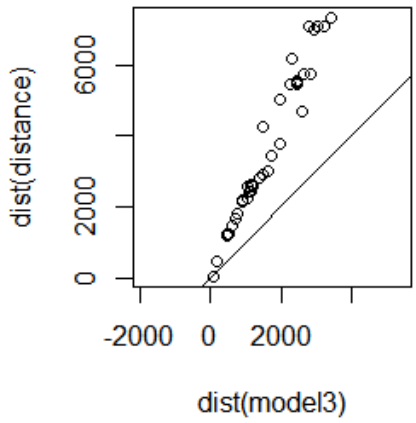


Again, I plot "$y = x$" graph and I found that the points are nearly on the line $y = x$. And the GOF for 2-dimensional model is 0.995. It is really high, indicating that a good fit.

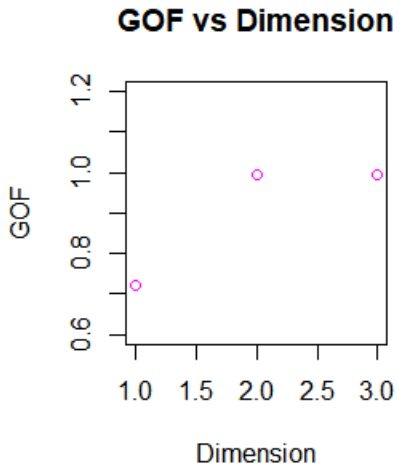


Lastly, I use 3-dimensional model. As before, I draw the histogram of this model and original distances vs this model. Even though the GOF value is 3-dimensional model is really high: it is 0.999, after I draw the line $y = x$, I found that the points deviate from the $y = x$ a lot, which may affect the final map.





The plot below shows GOF versus dimension. The GOF increase as the dimension increases and we observe that in these cities that are close to each other, the GOF value is really high.



I make a table for comparing the original and 2-dimensional model. Below is the difference between model2 and original distances. We can see that the difference is not that big.

3.76								
5.28	7.18							
-0.405	7.24	5.47						
4.92	15.32	-0.50	1.401					
8.89	1.20	3.64	10.19	3.69				
10.193	1.46	4.13	11.93	3.68	0.90			
2.80	5.509	0.0004	0.975	0.74	2.33	3.15		
0.02	2.77	-1.13	1.04	2.10	0.76	0.93	-0.95	
5.36	2.58	-0.6	5.83	1.79	-0.057	0.82	0.8	1.46

I finally draw the map of 2-dimensional model and 3-dimensional model. For 2-dimensional model, I notice that location are really close to the true relative location. However, in the real world, ShenZhen and Hong Kong are in the north of China.

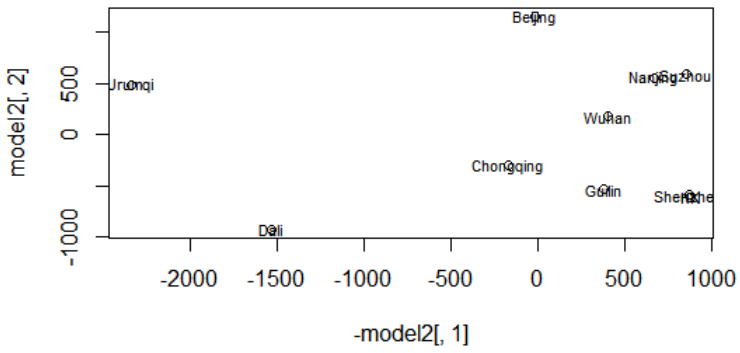
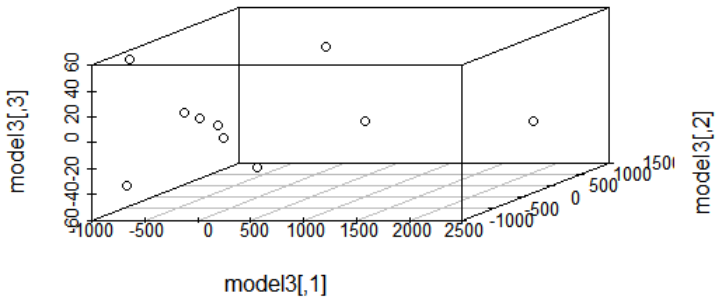
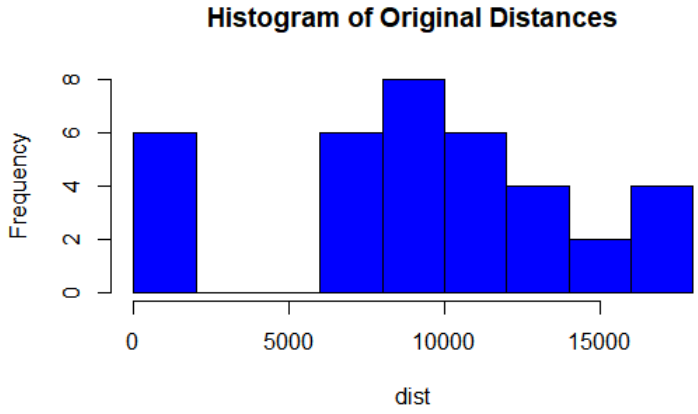


Figure 1: Map, k = 2

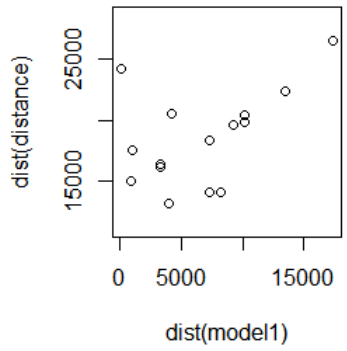


2.2 Data Set 2

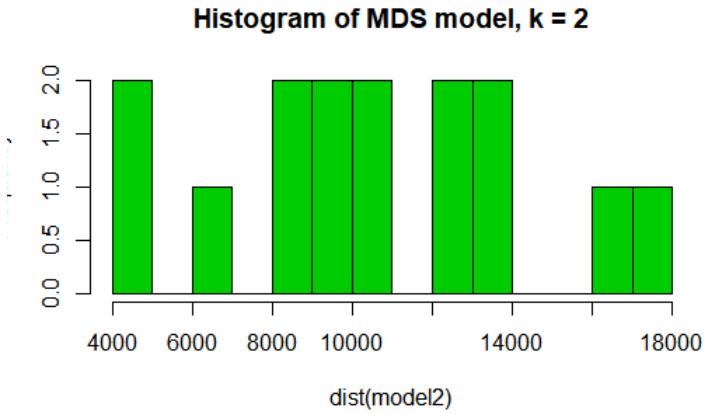
Here is the histogram of the original distances between six cities. We observe that most cities are far apart: more than 10000 km, since these six cities are in different continents.

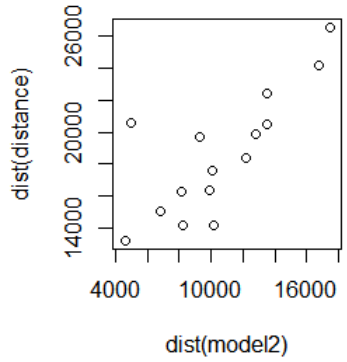


First is $k = 1$ dimensional model. The output is a list of 1-dimensional coordinates, I plot the result vs the original distances graph. We can see that the points are not close to the line $y = x$; they spread a lot. And the GOF value is only 0.339, indicating that it is not really a good fit.

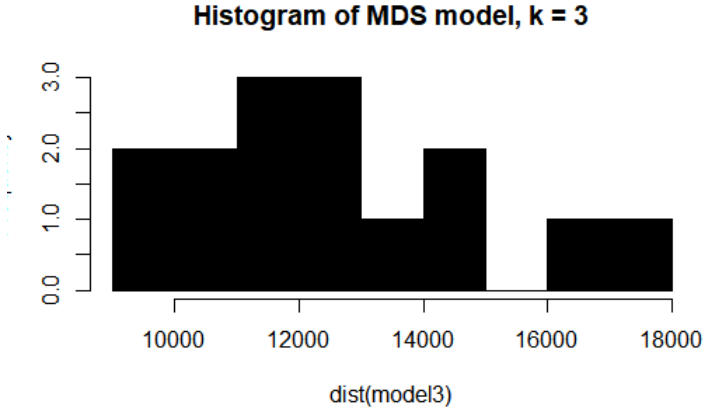


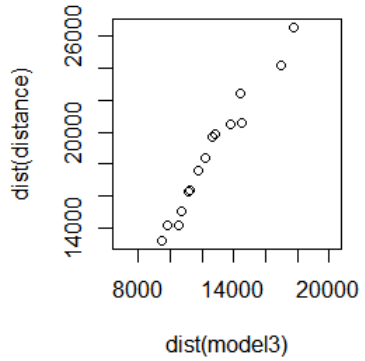
The second part is when k is 2. We will first look at the histogram. We can see that still most distances are above 10000 km. Due to my own choice of breaks for the histogram, it looks with more spread than the original distance histogram. For the "$y = x$" graph, we notice that the points are closer to line $y = x$ than case 1 while $k = 1$. And the GOF increases to 0.634. We will focus more on the case: $k = 3$ and draw the map



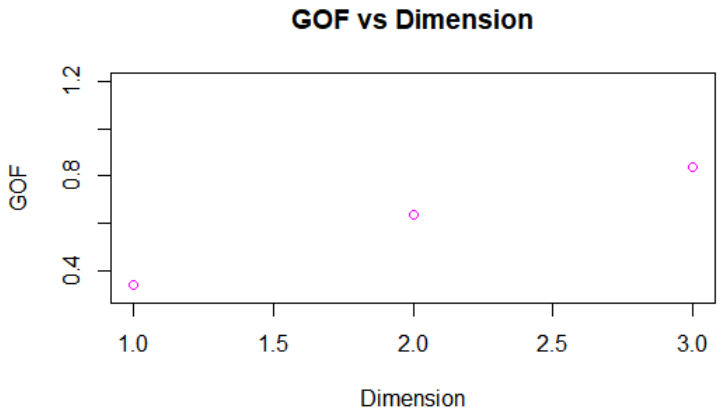


The last part of the analysis is when k is 3. We can check the histogram and $y = x$ graph. The GOF value is much better: 0.838. Even though the increment of dimension will increase GOF, 0.838 still points out that it is a good fit. We can see from the histogram that nearly all the distances are above 10000. In the $y = x$ graph, we can see that the points are really close to $y = x$, still existing a few outliers. This indicate that it is approaching our ideal situation that all the dost lie on $y = x$.





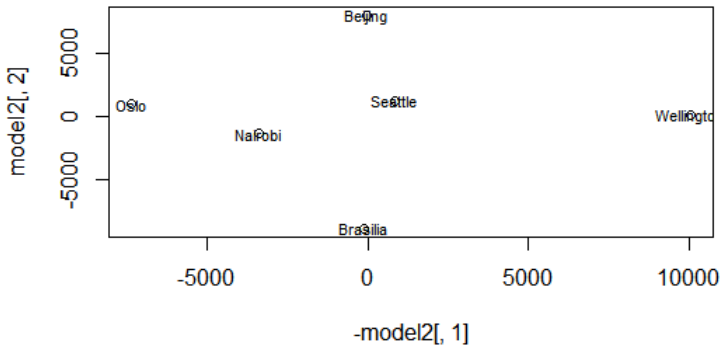
Next, we draw the relationship between dimension and GOF. It is a clear positive relationship. Now, it is time to see our map. I draw the map in 2-dimension and 3-dimension.

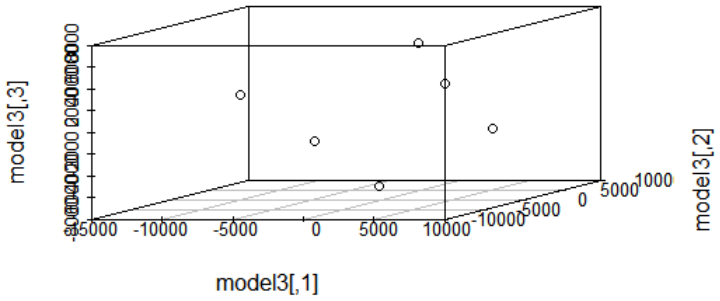


Also, here is the table for comparing the difference between model 3 and original distances. We can see that even in 3D, the difference is really big.

2049.38				
2044.003	745.14			
17.131	1484.12	1779.109		
1997.472	1001.14	8.578	1562.167	
3509.89	71.66	2287.332	2275.000	2485.2197

We can see that approximate location are correct, but for example, the relative location between Beijing and Seattle is kind of wrong. Seattle is in the east of Beijing: their latitude is almost the same. Even tough in 3D, the fit is not that good. I guess it is because the distance I choose are not straight-line distances. Since the cities are in different continents, distances are easily affected by some geographical features. Distances may not be Euclidean distances, but in R, the MDS algorithm assumes to use the Euclidean distances. I guess those may be the reasons that it is not a good fit even in 2D and 3D.





3 R code

3.1 Data Set 1

```
library(wordcloud)In
library(scatterplot3d)

# read data set and add column/row names
dir = "C:/Users/cuijy/Desktop"
distance <- read.csv(file.path(dir, "data.csv"), header = F)
row.names(distance) <- c("Beijing", "Dali", "Shenzhen", "HK",
                         "Urumqi", "Nanjing", "Suzhou", "Guilin", "Chongqing", "Wuhan")
colnames(distance) <- c("Beijing", "Dali", "Shenzhen", "HK",
                        "Urumqi", "Nanjing", "Suzhou", "Guilin", "Chongqing", "Wuhan")

dist = as.matrix(distance)
hist(dist, breaks = 10, main = "Histogram of Original Distances", col = 4)

model1 <- cmdscale(dist, k = 1)
plot(model1, asp = 1)
hist(dist(model1), breaks = 10, main = "Histogram of MDS model, k = 1", col = 2)
par(pty="s")
```

```
plot(dist(model1), dist(distance), asp = 1)
# how good is this model
gof_val = cmdscale(dist, k = 1, eig = T)$GOF # 0.72, 0.723

eig_val = eigen(dist)
plot(eig_val$values, ylim = c(min(eig_val$values), 20000), main = "Eigenvalues")

plot(dist, as.matrix(dist(model1)))

model2 <- cmdscale(dist, k = 2)
plot(model2, asp = 1, xlim = c(-1100, 2400), ylim = c(-1020, 1300))
hist(dist(model2), breaks = 10, main = "Histogram of MDS model, k = 2", col = 3)
plot(dist(model2), dist(distance), asp = 1)
# how good is this model
gof_val_2 = cmdscale(dist, k = 2, eig = T)$GOF # 0.995, 0.999

model3 <- cmdscale(dist, k = 3)
plot(model3, asp = 1)
hist(dist(model3), breaks = 10, main = "Histogram of MDS model, k = 3", col = 1)
plot(dist(model3), dist(distance), asp = 1)
# how good is this model
gof_val_3 = cmdscale(dist, k = 3, eig = T)$GOF # 0.996, 1

GOF <- c(gof_val[1], gof_val_2[1], gof_val_3[1])
plot(c(1, 2, 3), GOF, ylim = c(0.6, 1.2), col = 6, main = "GOF vs Dimension",
     xlab = "Dimension")

text(-model2[,1],model2[,2],c("Beijing", "Dali", "Shenzhen", "HK",
                              "Urumqi", "Nanjing", "Suzhou", "Guilin",
                              "Chongqing", "Wuhan"), cex=.7)
plot(-model2[,1], model2[,2])
table = dist(model2)-as.dist(distance)
scatterplot3d(model2)
```

3.2 Data Set 2

```
# read data set and add column/row names
dir = "C:/Users/cuijy/Desktop"
distance <- read.csv(file.path(dir, "data1.csv"), header = F)

colnames <- c("Beijing", "Wellington", "Nairobi", "Brasilia", "Seattle", "Oslo")

dist = as.matrix(distance)
hist(dist, breaks = 10, main = "Histogram of Original Distances", col = 4)

model1 <- cmdscale(dist, k = 1)
plot(model1, asp = 1)
hist(model1, main = "Histogram of MDS model, k = 1", col = 2)

par(pty="s")
plot(dist(model1), dist(distance), asp = 1)
# how good is this model
gof_val = cmdscale(dist, k = 1, eig = T)$GOF # 0.339, 0.405
```

```
model2 <- cmdscale(dist, k = 2)
plot(model2, asp = 1, xlim = c(-1100, 2400), ylim = c(-1020, 1300))
hist(dist(model2), breaks = 10, main = "Histogram of MDS model, k = 2", col = 3)
plot(dist(model2), dist(distance), asp = 1)
abline(x = 0, y = 1)
# how good is this model
gof_val_2 = cmdscale(dist, k = 2, eig = T)$GOF # 0.634, 0.757

model3 <- cmdscale(dist, k = 3)
plot(model3, asp = 1)
hist(dist(model3), breaks = 10, main = "Histogram of MDS model, k = 3", col = 1)
plot(dist(model3), dist(distance), asp = 1)
# how good is this model
gof_val_3 = cmdscale(dist, k = 3, eig = T)$GOF # 0.838, 1

GOF <- c(gof_val[1], gof_val_2[1], gof_val_3[1])
plot(c(1, 2, 3), GOF, ylim = c(0.3, 1.2), col = 6, main = "GOF vs Dimension",
     xlab = "Dimension")

text(-model2[,1],model2[,2],c("Beijing", "Wellington", "Nairobi",
                              "Brasilia", "Seattle", "Oslo"), cex=.7)
plot(-model2[,1], model2[,2])

table = dist(model3)-as.dist(distance)
scatterplot3d(model3)
```