math381 hw7

Wenhao Ma

March 2023

1 Introduction

Given a data set of 1970 crime rates for American cities as of 1970. This is a list of cities and the number of crimes per 100,000 population:

"City"	"Murder"	"Rape"	"Robbery"	"Assault"	"Burglary"	"Larceny"	"Auto"
"Atlanta"	16.5	24.8	106	147	1112	905	494
"Boston"	4.2	13.3	122	90	982	669	954
"Chicago"	11.6	24.7	340	242	808	609	645
"Dallas"	18.1	34.2	184	293	1668	901	602
"Denver"	6.9	41.5	173	191	1534	1368	780
"Detroit"	13.0	35.7	477	220	1566	1183	788
"Hartford"	2.5	8.8	68	103	1017	724	468
"Honolulu"	3.6	12.7	42	28	1457	1102	637
"Houston"	16.8	26.6	289	186	1509	787	697
"Kansas City"	10.8	43.2	255	226	1494	955	765
"Los Angeles"	9.7	51.8	286	355	1902	1386	862
"New Orleans"	10.3	39.7	266	283	1056	1036	776
"New York"	9.4	19.4	522	267	1674	1392	848
"Portland"	5.0	23.0	157	144	1530	1281	488
"Tucson"	5.1	22.9	85	148	1206	756	483
"Washington"	12.5	27.6	524	217	1496	1003	739

Figure 1: 1970 crime rates for American cities

Here, Murder is the murder rate. Rape is the rape rate. Robbery is the robbery rate. Assault is the assault rate. Burglary is the burglary rate. Larceny is the larceny rate. Auto is the auto theft rate.
Now, I want to do multidimensional scaling to those data and see if I can find something interesting.

2 Analysis

First, I need to adjust the columns of the input data so that they are all comparable. This is to make sure that the scale of each dimension of the vectors does not overly affect the distance calculations. I will use two ways of normalization

to see which one is better.

Standard normalization

This one is finding the mean and standard deviation of each column, and then subtracting the mean and dividing by the standard deviation in each column. This will transform our data into a data set in which each column has a mean of 0 and a standard deviation of 1. After I standardize, I got the following table:

City	Murder	Rape	Robbery	Assault	Burglary	Larceny	Auto
Atlanta	1.37486233	-0.27521745	-0.87456350	-0.58319087	-0.8644072	-0.376787139	-1.29349986
Boston	-1.13044236	-1.22888998	-0.77279611	-1.25815289	-1.2905667	-1.278973717	1.75587841
Chicago	0.37681412	-0.28351026	0.61378457	0.54174583	-1.8609648	-1.508343187	-0.29250830
Dallas	1.70075563	0.50430618	-0.37844748	1.14565922	0.9582442	-0.392078436	-0.57755887
Denver	-0.58049743	1.10968091	-0.44841256	-0.06216755	0.5189721	1.393180599	0.60241793
Detroit	0.66197075	0.62869825	1.48516784	0.28123418	0.6238729	0.685958069	0.65545060
Hartford	-1.47670399	-1.60206619	-1.11626105	-1.10421419	-1.1758315	-1.068718371	-1.46585602
Honolulu	-1.25265235	-1.27864681	-1.28163306	-1.99232211	0.2665546	0.376309285	-0.34554096
Houston	1.43596732	-0.12594697	0.28940101	-0.12137475	0.4370184	-0.827880428	0.05220403
Kansas City	0.21386747	1.25065859	0.07314531	0.35228281	0.3878461	-0.185645914	0.50298169
Los Angeles	-0.01018417	1.96383979	0.27031963	1.87982844	1.7253314	1.461991440	1.14600276
New Orleans	0.11202582	0.96041043	0.14311039	1.02724483	-1.0479836	0.124002869	0.57590160
New York	-0.07128916	-0.72302890	1.77138862	0.83778181	0.9779131	1.484928387	1.05319559
Portland	-0.96749572	-0.42448793	-0.55017995	-0.61871519	0.5058595	1.060594869	-1.33327436
Tucson	-0.94712738	-0.43278074	-1.00813320	-0.57134943	-0.5562611	-0.946387987	-1.36641977
Washington	0.56012910	-0.04301892	1.78410954	0.24570986	0.3944024	-0.002150339	0.33062552

Figure 2: standardized 1970 crime rates for American cities

Then calculate our distance matrix from that using classic Euclidean distance and we get(It is a 16 by 16 matrix. Here I only show the first 5 rows and column):

	1	2	3	4	5
1	0.000000	4.528795	2.976549	2.982574	4.117624
2	4.528795	0.000000	3.839890	5.681261	4.666264
3	2.976549	3.839890	0.000000	3.854193	4.688495
4	2.982574	5.681261	3.854193	0.000000	3.673346
5	4.117624	4.666264	4.688495	3.673346	0.000000

Figure 3: Euclidean distance between cities(First 5 rows and columns)

The numbers represent the cities in the order I showed before. Number 1 represents Atlanta, number 2 represents Boston, number 3 represents Chicago, etc. And [2,1] represents the distance between Boston and Atlanta.

Then I'm going to use MDS to create low-dimensional models with cmdscale command in R. I plot the 1d and 2d model here:

1 Dimensional

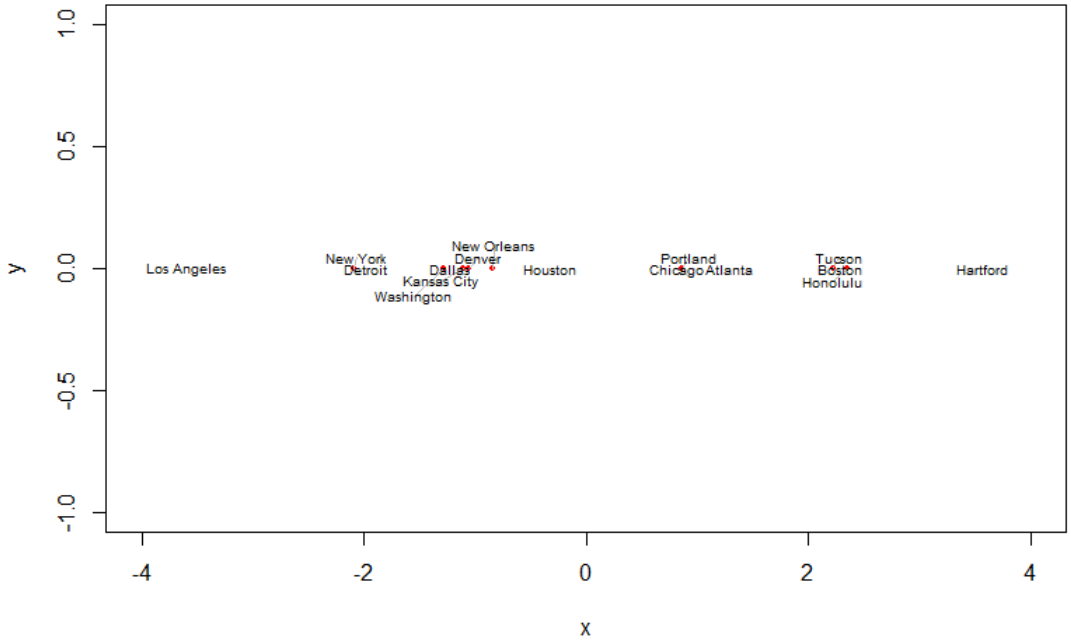

Figure 4: 1 Dimensional Model

2 Dimensional

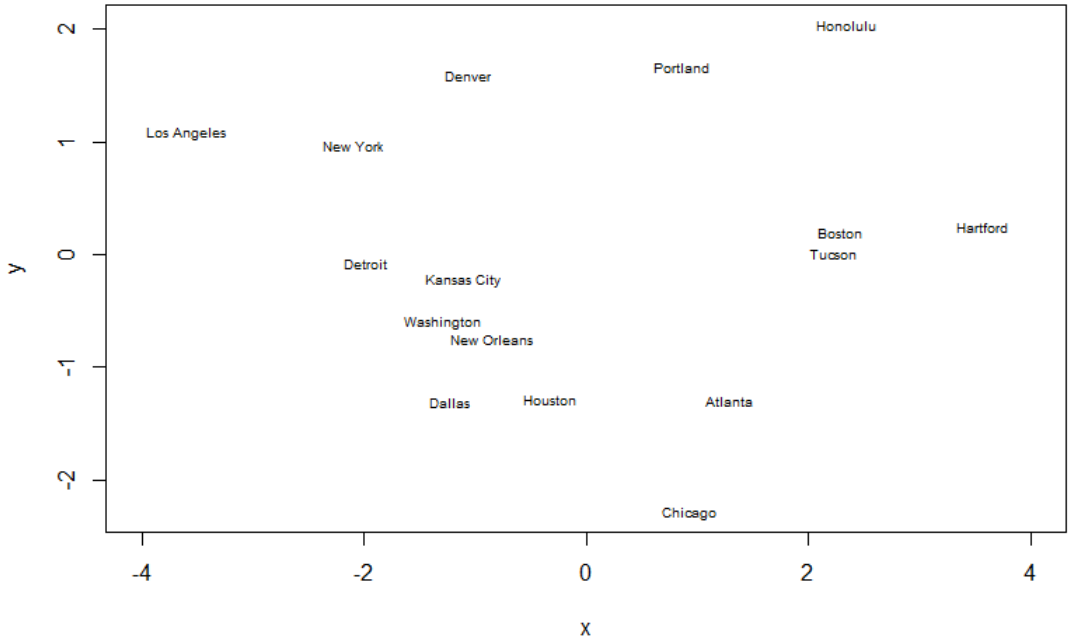

Figure 5: 2 Dimensional Model

How good is this model?

First, I calculate the eigenvalue and plot them from largest to smallest, we see this:

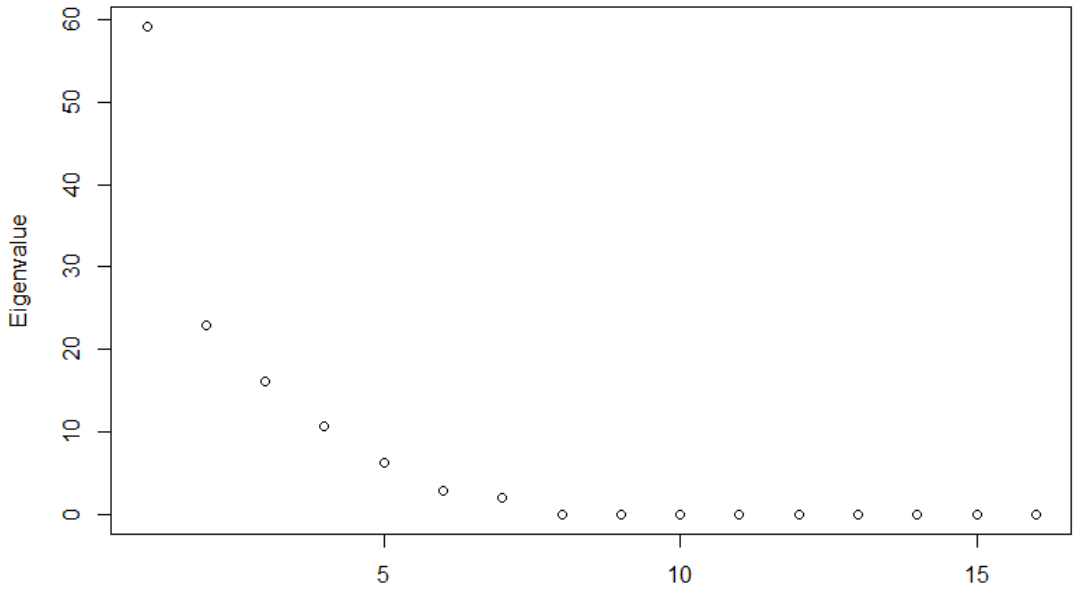

Figure 6: Eigen Value

In this plot, I find that there are in total 7 non-zero eigenvalues. This corresponds to the fact that our data actually fits perfectly into 7-dimensional space. And the first value is much higher than other values. The next value is not tiny. Therefore, this suggests that the 2-dimensional model captures some information of the data but is not perfect.

Then, I calculate GOF values for each model:

	GOF
Model1	0.4930817
Model2	0.683463
Model3	0.8177821

Figure 7: GOF Value

The GOF values increase when the dimensions of the models increase. When I calculate the GOF values in R, it gives me two same values. This is because R uses eigenvalues to calculate the goodness of fit, so it has a different way to deal with the negative eigenvalue. (If eigenvalues are 0 then R will use 0 instead

of the absolute value) However, in my case, there is no negative eigenvalue, so I get 2 same values.

Next, I do some comparison of distance matrix of each model to the input distance matrix:

Method 1. The sum of the absolute difference of entries

Method 2. The sum of the square of the difference of entries

Method 3. The maximum of absolute difference of entries

	Method 1	Method 2	Method 3
Model1	364.2379	754.8629	4.438501
Model2	204.299	278.329	3.442948
Model3	112.2394	85.75839	2.224685

Figure 8: Comparison between the original matrix and the model matrix

We can see that with the increase of dimensions, the difference between the input distance and the distance of the model will decrease.

I also plot the distances in the model versus the input distances(The black line is y=x):

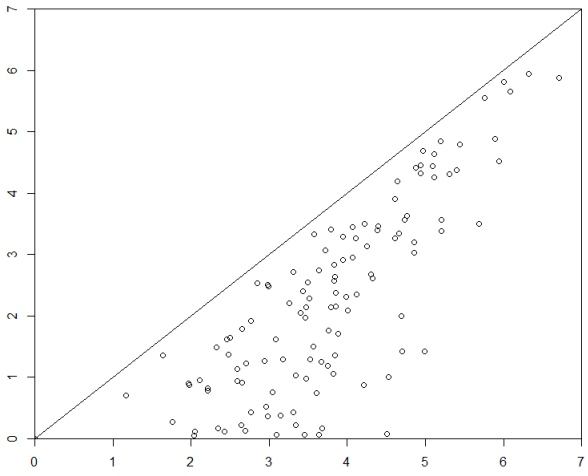

Figure 9: the distances in the model1(x-axis) versus the input distances(y-axis)

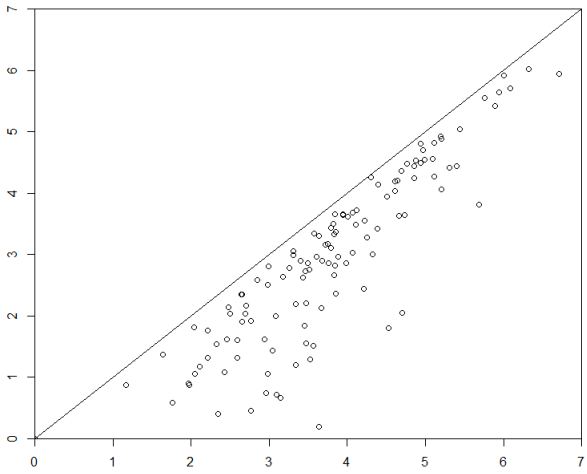

Figure 10: the distances in the model2(x-axis) versus the input distances(y-axis)

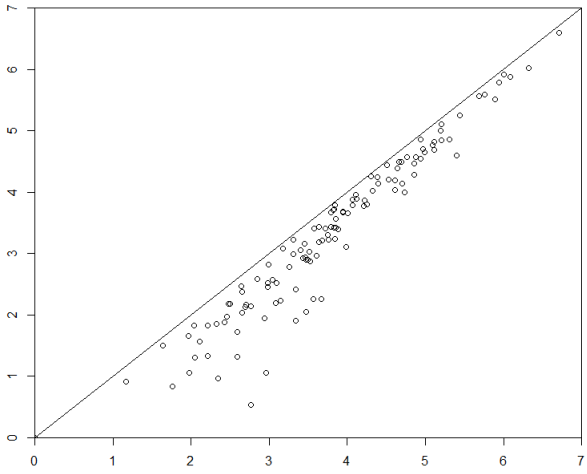

Figure 11: the distances in the model3(x-axis) versus the input distances(y-axis)

These 3 graphs show the similar result we have above: with the increase of dimensions, the difference between the input distance and the distance of the model will decrease, which means that the model will become better.

Another way of normalization

Another method is to find the min and max value in each column, and then map each column entry x to $\frac{x-min}{max-min}$. This will convert the data set into a new data set in which each column has a minimum of 0 and a maximum of 1. By this way, I got the following table:

	City	Murder	Rape	Robbery	Assault	Burglary	Larceny	Auto
1	Atlanta	0.89743590	0.37209302	0.13278008	0.3639144	0.2778793	0.37803321	0.05349794
2	Boston	0.10897436	0.10465116	0.16597510	0.1896024	0.1590494	0.07662835	1.00000000
3	Chicago	0.58333333	0.36976744	0.61825726	0.6544343	0.0000000	0.00000000	0.36419753
4	Dallas	1.00000000	0.59069767	0.29460581	0.8103976	0.7861060	0.37292465	0.27572016
5	Denver	0.28205128	0.76046512	0.27178423	0.4984709	0.6636197	0.96934866	0.64197531
6	Detroit	0.67307692	0.62558140	0.90248963	0.5871560	0.6928702	0.73307791	0.65843621
7	Hartford	0.00000000	0.00000000	0.05394191	0.2293578	0.1910420	0.14687101	0.00000000
8	Honolulu	0.07051282	0.09069767	0.00000000	0.0000000	0.5932358	0.62962963	0.34773663
9	Houston	0.91666667	0.41395349	0.51244813	0.4831804	0.6407678	0.22733078	0.47119342
10	Kansas City	0.53205128	0.80000000	0.44190871	0.6055046	0.6270567	0.44189017	0.61111111
11	Los Angeles	0.46153846	1.00000000	0.50622407	1.0000000	1.0000000	0.99233716	0.81069959
12	New Orleans	0.50000000	0.71860465	0.46473029	0.7798165	0.2266910	0.54533844	0.63374486
13	New York	0.44230769	0.24651163	0.99585062	0.7308869	0.7915905	1.00000000	0.78189300
14	Portland	0.16025641	0.33023256	0.23858921	0.3547401	0.6599634	0.85823755	0.04115226
15	Tucson	0.16666667	0.32790698	0.08921162	0.3669725	0.3638026	0.18773946	0.03086420
16	Washington	0.64102564	0.43720930	1.00000000	0.5779817	0.6288848	0.50319285	0.55761317

Figure 12: 1970 crime rates for American cities re-range to 0-1

Then calculate our distance matrix from that using classic Euclidean distance and we get(It is a 16 by 16 matrix. Here I only show the first 5 rows and column):

	1	2	3	4	5
1	0.0000000	1.404285	0.9167745	0.8222237	1.2701919
2	1.4042852	0.000000	1.1470929	1.6691052	1.4127442
3	0.9167745	1.147093	0.0000000	1.1295308	1.4520811
4	0.8222237	1.669105	1.1295308	0.0000000	1.1449056
5	1.2701919	1.412744	1.4520811	1.1449056	0.0000000

Figure 13: Euclidean distance between cities(First 5 rows and columns)

The numbers represent the cities in the order I showed before. Number 1 represents Atlanta, number 2 represents Boston, number 3 represents Chicago, etc. And [2,1] represents the distance between Boston and Atlanta.

Then I'm going to use MDS to create low-dimensional models with cmdscale command in R. I plot the 1d and 2d model here:

1 Dimensional

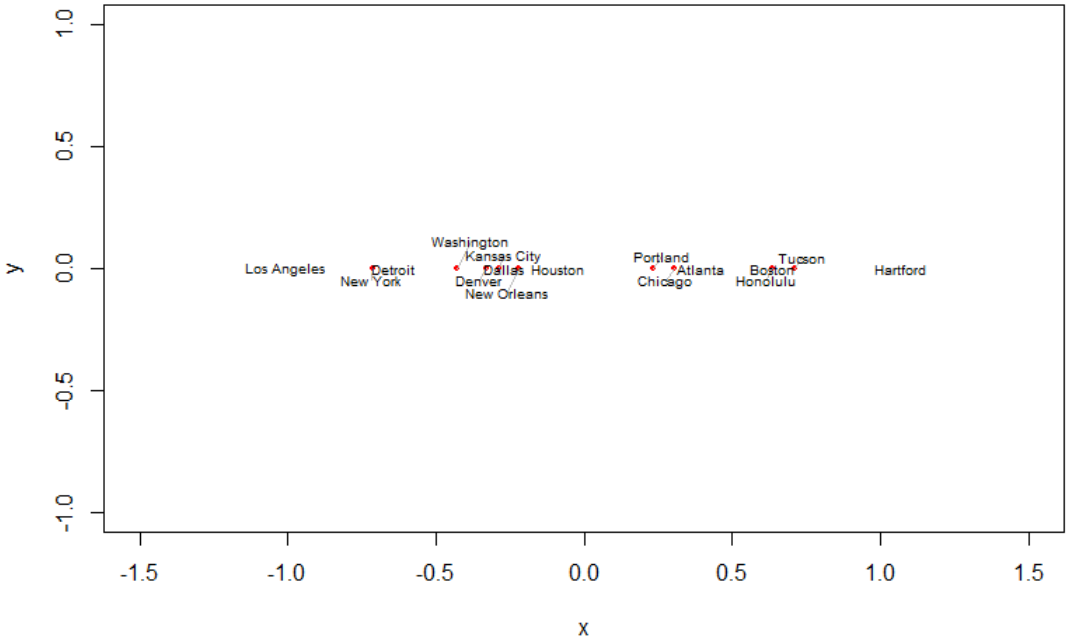

Figure 14: 1 Dimensional Model

2 Dimensional

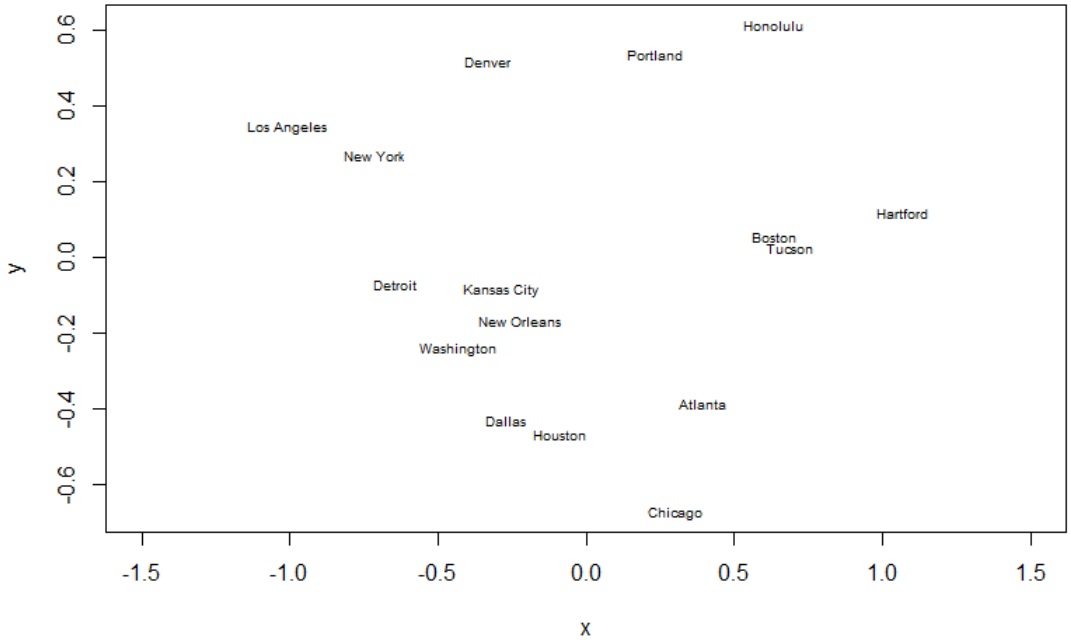

Figure 15: 2 Dimensional Model

Comparing these 2 graphs with Figure 4 and Figure 5, we can see they are very similar visually.

How good is this model?

First, I calculate GOF values for each model:

	GOF
Model1	0.4768181
Model2	0.6827583
Model3	0.8233565

Figure 16: GOF Value

Comparing this figure with Figure 7, I find that their GOF values don't have big differences. So for this data set, different normalization methods do not have big differences

Next, I do some comparison of the distance matrix of each model to the input distance matrix:

Method 1. The sum of the absolute difference of entries

Method 2. The sum of the square of the difference of entries

Method 3. The maximum of absolute difference of entries

	Method 1	Method 2	Method 3
Model1	111.6709	70.79987	1.3195
Model2	60.72897	24.80452	1.033764
Model3	32.21939	6.848707	0.581737

Figure 17: Comparison between the original matrix and the model matrix

We can see that with the increase of dimensions, the difference between the input distance and the distance of the model will decrease. However, comparing this with Figure 8, we find that the values decrease a lot. That's because the scale of the model also decreases. By comparing Figure 17 and Figure 8, we can see that values are just scaled.

I also plot the distances in the model versus the input distances(The black line is y=x):

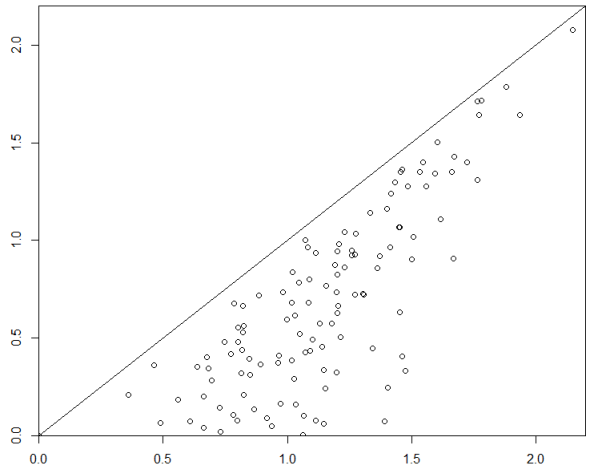

Figure 18: the distances in the model1(x-axis) versus the input distances(y-axis)

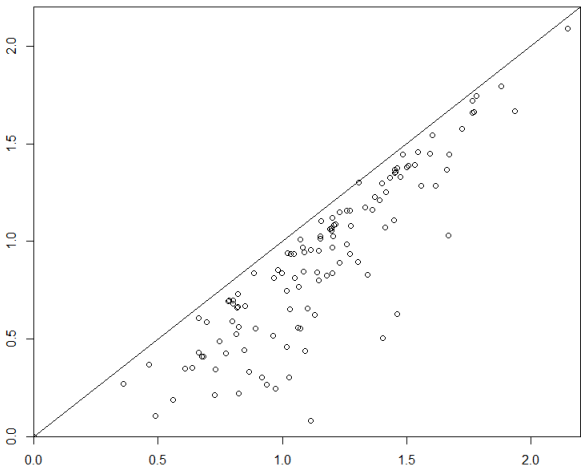

Figure 19: the distances in the model2(x-axis) versus the input distances(y-axis)

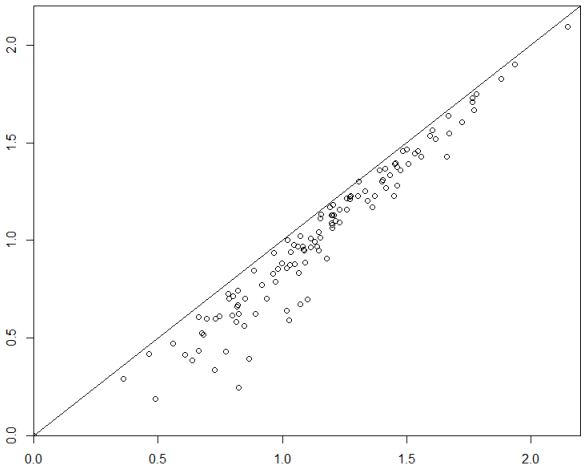

Figure 20: the distances in the model3(x-axis) versus the input distances(y-axis)

These 3 graphs show the similar result we have above: with the increase of dimensions, the difference between the input distance and the distance of the model will decrease, which means that the model will become better.

3 Conclusion

By the above analysis, we find that the models built and the goodness of the models are similar in 2 different methods of normalization. In the following analysis, I will use the model generated by the normalization method that maps all the data onto 0-1.

In the graph, Los Angeles and Hartford are farthest away from each other horizontally; Honolulu and Chicago are furthest away from each other vertically.

"City"	"Murder"	"Rape"	"Robbery"	"Assault"	"Burglary"	"Larceny"	"Auto"
"Hartford"	2.5	8.8	68	103	1017	724	468
"Los Angeles"	9.7	51.8	286	355	1902	1386	862

Figure 21: Los Angeles and Hartford

"City"	"Murder"	"Rape"	"Robbery"	"Assault"	"Burglary"	"Larceny"	"Auto"
"Chicago"	11.6	24.7	340	242	808	609	645
"Honolulu"	3.6	12.7	42	28	1457	1102	637

Figure 22: Chicago and Honolulu

I also calculate the correlation coefficient of the x,y coordinates versus the rate of the different types of crimes. This is what I get:

	x coordinate	y coordinate
Murder	-0.5011989	-0.7372762
Rape	-0.7720286	-0.06647641
Robbery	-0.7459283	-0.2696631
Assault	-0.8204511	-0.3076262
Burglary	-0.7294403	0.397781
Larceny	-0.6764707	0.6660193
Auto	-0.6044029	0.103475

Figure 23: x,y coordinates versus the rate of the different types of crimes

I also sum the rates of crimes for each city and plot that against the x-coordinate of the model:

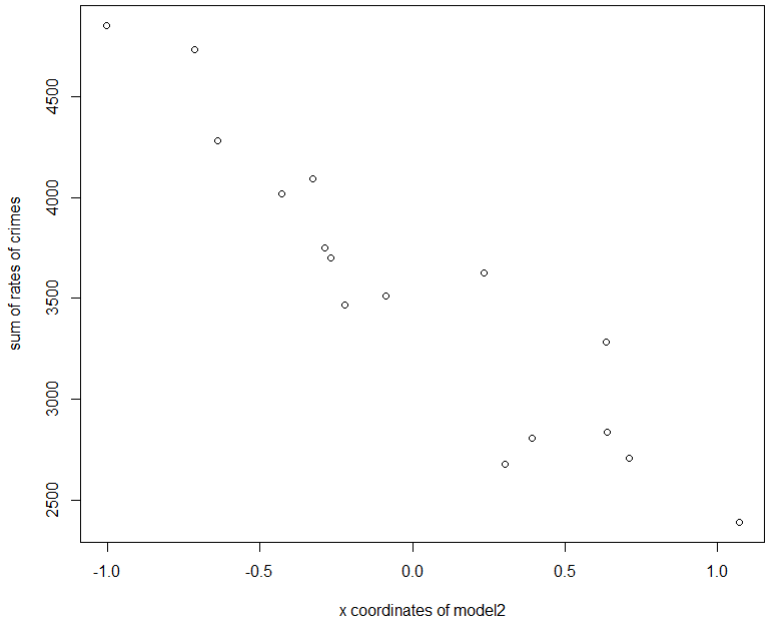

Figure 24: Sum of rates of crimes versus x-coordinates of the model

In Figure 23, I find that x coordinates have a high negative relationship with almost all types of crimes and y coordinates have no relationship with almost all types of crimes. And Figure 24 shows that for a city, when the sum of its rates of crimes increases, it will be located more left in the model; And when its rates of crimes increases, it will be located at right of the model.

In Figure 21, I find that Hartford and Los Angeles are different in all types of crimes. Los Angeles has a lot of crimes but Hartford is relatively safe. For Chicago and Honolulu, in these two cities, their crimes has different composition. Chicago has a high rate of murder, pare, robbery and assault but a low rate of burglary, larceny and auto, which is opposite from Honolulu.

Therefore, I conclude that x-axis evaluates the sum of rates the crimes and the y-axis evaluates the composition of types of crimes. In other word, if the number of crimes happening in two cities are different, they will be far a part on x-coordinates; However, if the proportion of each type of crime relative to the total number of crimes are different for 2 cities, they will be far away on y-coordinates.

4 Appendix

https://github.com/aaamwh/Math-381-hw7.git