

Bob Lin
Math381
Homework5

Here, I will explore the distance of several city I like in Asian and label them as cities that are close together which means the largest distance among each of pair will less than 5000 km. I first collect a chart containing the distance information between them as following (the distance is the air distance):

City	BeiJing	ShenZhen	ShangHai	Bangkok	Seoul	Tykyo
BeiJing	0	2275.6	1084	3301	954	2100
ShenZhen	2275.6	0	1215	1698	2082	2873
ShangHai	1084	1215	0	2886	868	1765
Bangkok	3301	1698	2886	0	3654	4620
Seoul	954	2082	868	3654	0	1160
Tykyo	2100	2873	1765	4620	1160	0

To better analyze the model, I also collect a chart containing the distance information among some “far away” city which has at least one distance greater than 5000, and the chart is following:

City	Paris	Qingdao	Urumqi	NewYork	Moscow	Seattle
Paris	0	8780	8229	5840	2477	8044
Qingdao	8780	0	3519	11321	6352	8702
Urumqi	8229	3519	0	10441	3736	9457
NewYork	5840	11321	10441	0	7511	3800
Moscow	2477	6352	3736	7511	0	8404
Seattle	8044	8702	9457	3800	8404	0

To see the 1,2,3 dimensional models, I will use R to apply MDP(multidimensional scale) to this distance matrix, below is the code:

```
distances <- read.csv(file="381matrix.csv",
                      head=FALSE,sep=",")
model1 <- cmdscale(distances,k=1)
plot(model2,asp=1,
     ylab="",xlab="")
```

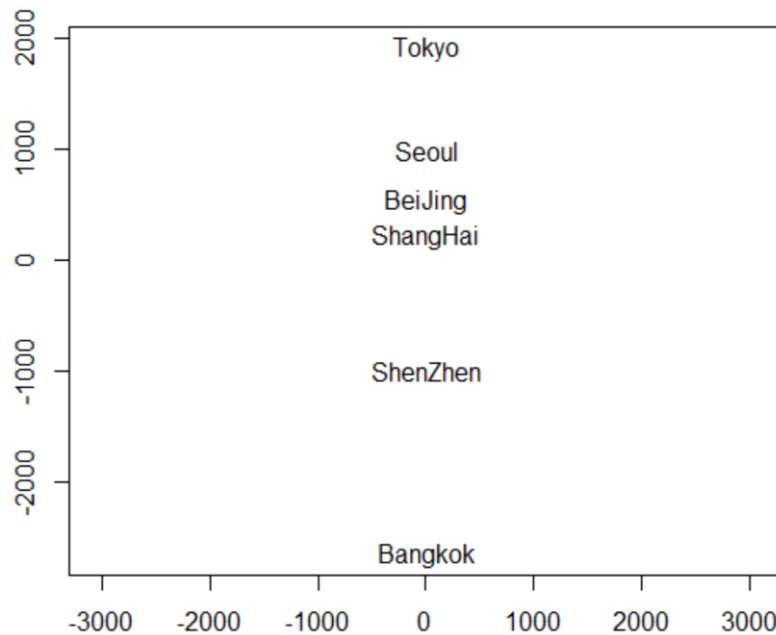
Also I use following code to generate the text plot which is clear to see the city position:

```
distplusnames <- read.csv(file="381name.csv",head=TRUE,sep=",")
text(model1[,1],
```

cex=1)

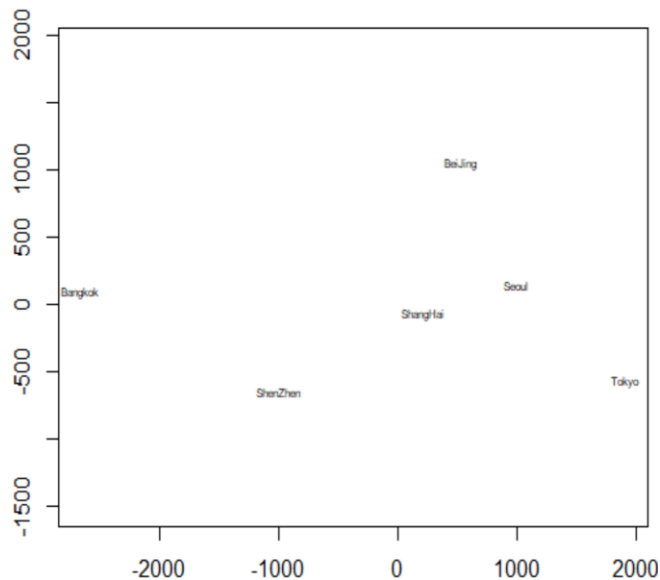
This creates the following textplot with name in each position:

The plot is a 1D model of cities close together:

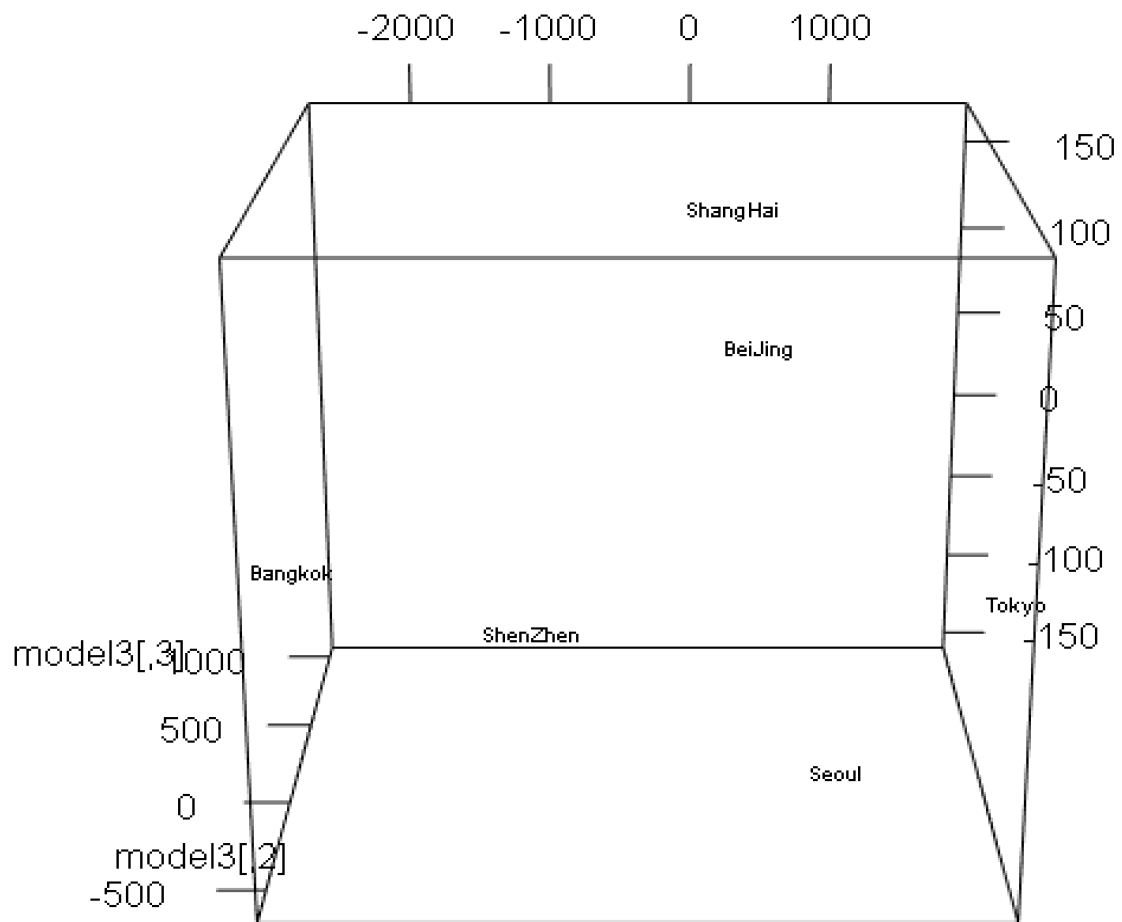


The city that far away from other is Bangkok, and others seem to be close.

Then I change the k of the code in the “cmdscale” part to 2 and 3 to creates 1 and 3-dimensional scale. Then it is a 2D and 3D model for us to see the cities that are close together:

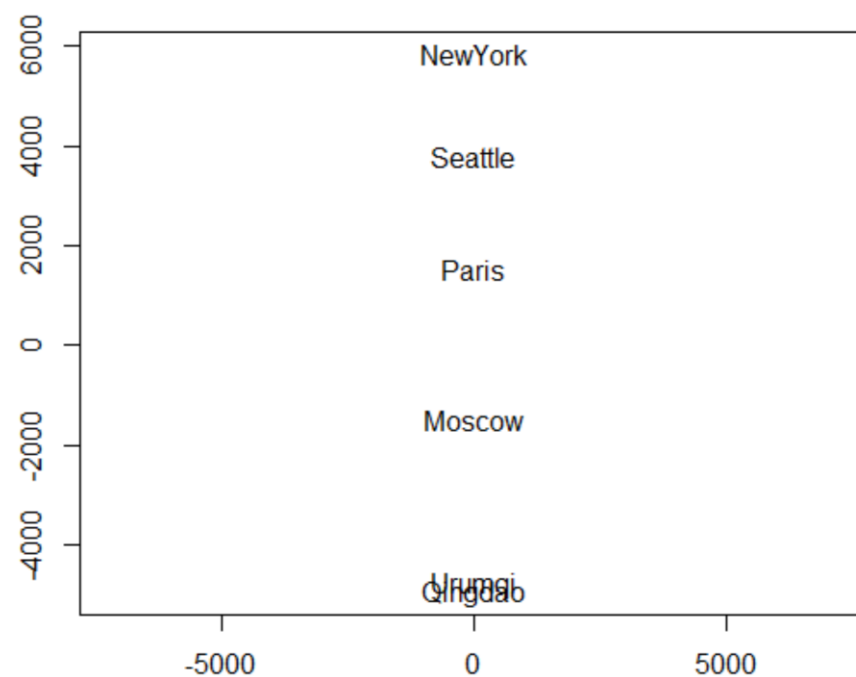


It is seems that BeiJing, ShangHai, and Seoul are pretty close to each other.

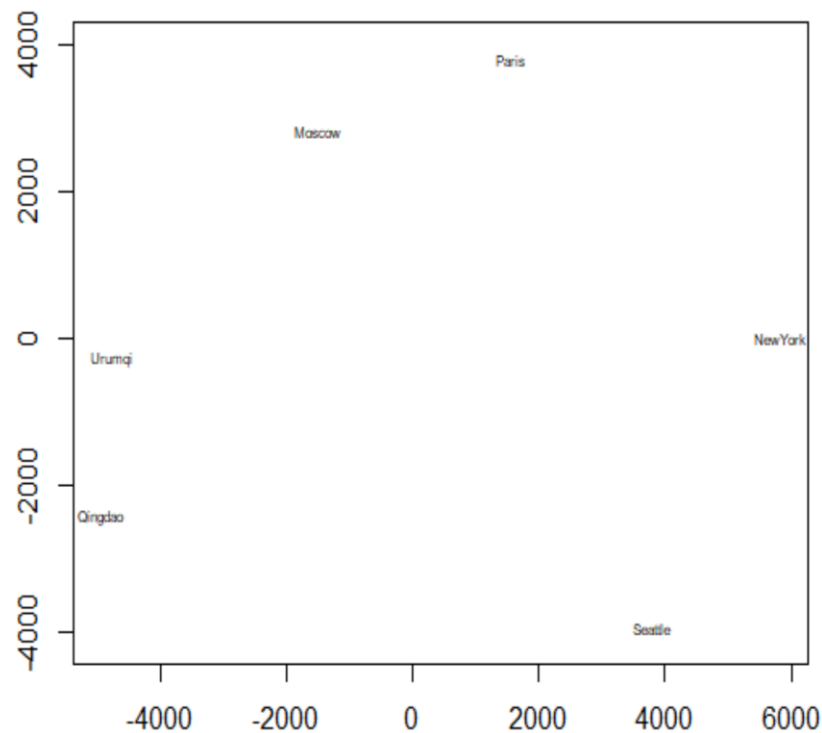


It is not so clear, to see, but I see that Bangkok and Tokyo has greatest distance

For the far away cities I listed above, I did the same thing, and below are the plots:
The plot with text in 1-dimensional:

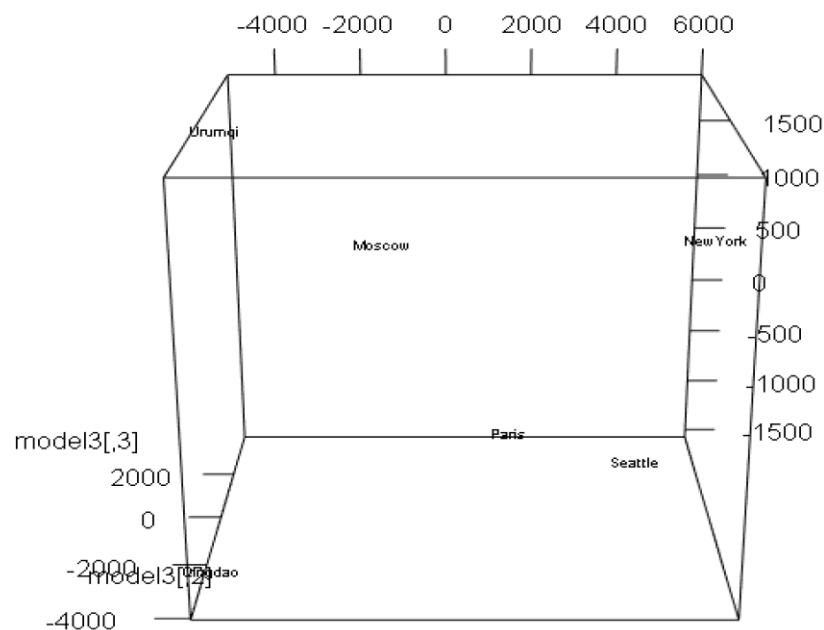


I see that Urumqi and Qingdao are close
The plot with text in 2-dimensional:



I see that Seattle and New York are far away from other, and I think this is because they are cities from America

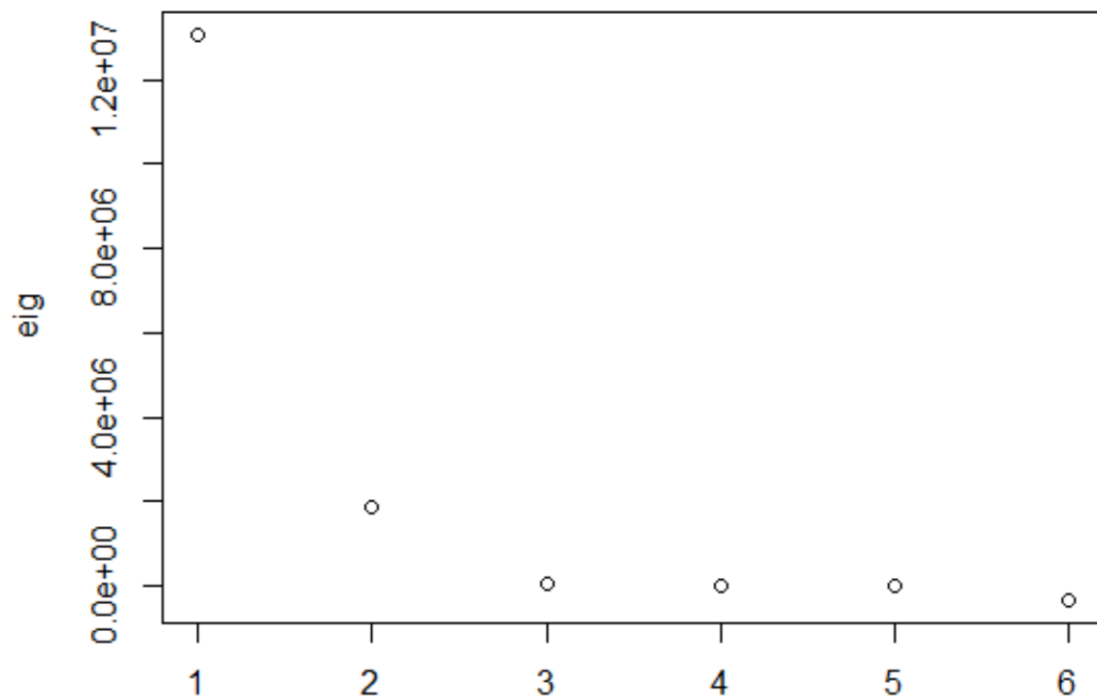
The plot with text in 3-dimensional:



I see that New York and Seattle are still close.
To see how good the models are, I will investigate the eigenvalue for them from the

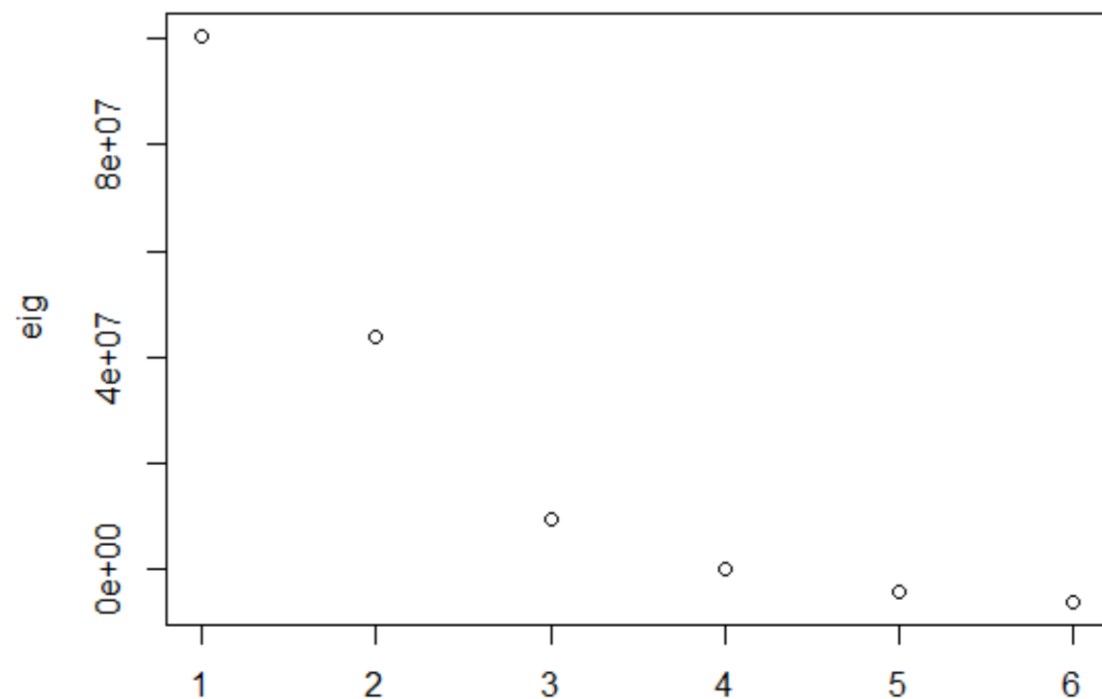
largest to smallest:

Eigenvalue for cities that are close together:



In the plot, I see that the first eigen value seems to be larger than other, and the third to sixth seem to be close to 0

Eigenvalue for cities that are far away:

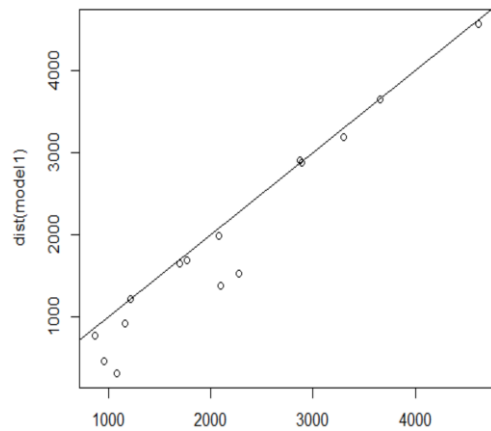


In the plot, the last two eigenvalues seem to be zero

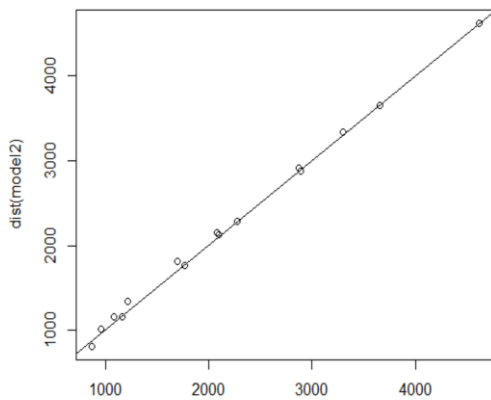
We see that about the first 3 are clearly non-zero, which means that 20-dimensional space fits the data best. Also, because first two eigenvalues are not small, 1-dimensional model and 2-dimensional model will not fit them perfectly.

Also, there is another way to see how good each model is, which is comparing the distance in model with original distance in plot. The plot is shown below:

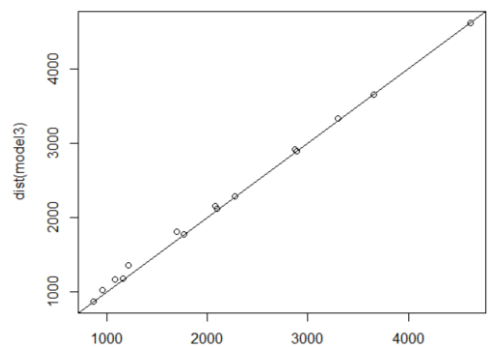
Original distance vs distance in 1,2,3-dimensional model of close together cities:



I see that the original number less than 3000 is disarray

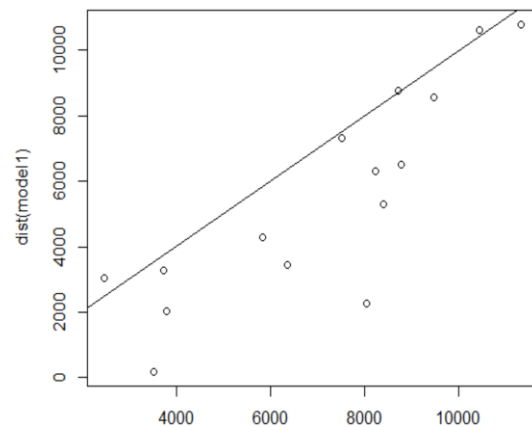


I see that all points seem to form a line

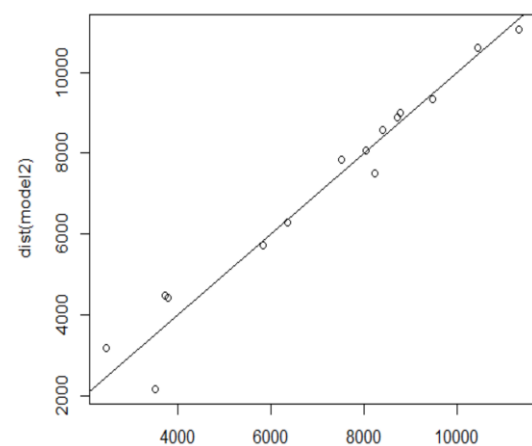


This plot is like the model2

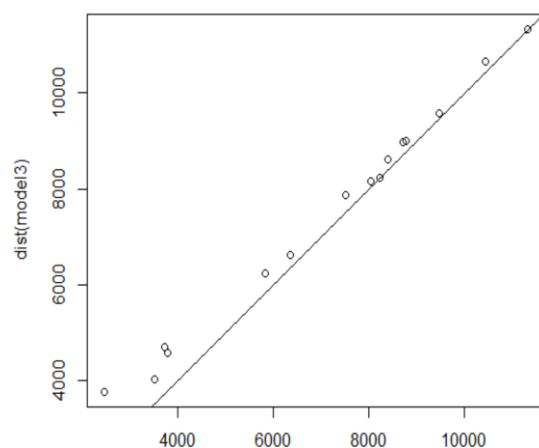
Original distance vs distance in model among cities that are far away:



I see that all the points are scattering



I see that most of points near the line $y = x$



I see that all points almost lie in a line.

The most ideal situation is all the dots lies in $y = x$. From the plots above, I see that the points almost lie in a line in the model3 vs original distance for the cities far away, and I see that the points almost lie in a line in the model2 vs original distance and model3 vs original distance for the cities close together. Thus, the model2 and model3 of cities that are close together is nice, and the model3 of cities that are far away is also nice.

Also, I compare all the distance model and the input distance below:

The table includes the difference between model distance of model1 and input distances for cities that are close together:

	BJ	SZ	SH	BK	SE
SZ	741.364064				
SH	767.514214	-2.750150			
BK	107.461557	38.697493	8.947343		
SE	492.276436	86.040500	89.790650	-1.262007	
TO	718.358668	-42.877267	66.872882	44.820226	240.082232

The table includes the difference between model distance of model2 and input distances for cities that are close together:

	BJ	SZ	SH	BK	SE
SZ	-7.268472				
SH	-78.337387	-130.410206			
BK	-31.797270	-117.658583	4.214980		
SE	-57.446910	-64.812401	59.694876	-1.652445	
TO	-23.945938	-43.901924	-3.722896	-2.511024	-4.020633

The table includes the difference between model distance of model3 and input distances for cities that are close together:

	V1	V2	V3	V4	V5
V2	-7.614751				
V3	-86.247411	-141.820925			
V4	-32.450355	-117.847873	-2.843107		
V5	-74.487873	-69.817720	-2.185030	-3.636750	
V6	-24.171112	-43.915305	-11.566995	-2.643932	-14.363676

We see that the difference become smaller in model2 and model3, but model2 seems perform better

The table includes the difference between model distance of model1 and input distances for cities that are far away:

	Pa	QD	UR	NY	MO
QD	2281.13267				
UR	1915.84609	3333.28658			
NY	1547.97708	530.10975	-164.17683		
MO	-557.50706	2887.63973	457.35315	184.47002	
SE	5786.03954	-54.82779	885.88563	1765.93753	3111.53248

The table includes the difference between model distance of model2 and input distances for cities that are far away:

	Pa	QD	UR	NY	MO
QD	-229.94602				
UR	719.32256	1337.43911			
NY	103.45226	258.92326	-167.38171		
MO	-711.73082	52.94948	-767.35492	-341.82034	
SE	-21.89279	-182.86006	130.64201	-631.66041	-184.25539

The table includes the difference between model distance of model3 and input distances for cities that are far away:

	Pa	QD	UR	NY	MO
QD	-229.9771911				
UR	-0.6211164	-512.2202122			
NY	-390.3061867	-10.1101829	-208.5439229		
MO	-1293.9027732	-268.0991800	-966.3091503	-352.9323226	
SE	-116.1432734	-271.8079426	-109.3989491	-789.7247958	-219.26301

I see that model2 has the lowest difference, so this is better than other two. And all data seems to have to greater difference in model3 like -189.25539 become -219.26301

Because the data is limited, the histogram is not an ideal choose for analyzing the model. But there is more I can do:

GOF is a property of the model from 0 to 1 with a value of 1 means a perfect fit. There is an interested thing that the GOF of the model will positively increases with the dimension until it reaches the dimension of the data. So, I calculate the GOF in 1-3 dimensions to see how GOF will change with dimension:

For cities that are close together:

1-Dimension: GOF = 0.8513311 0.8718616
 2-Dimension: GOF = 0.9729172 0.9963798
 3-Dimension: GOF = 0.9764522 1.0000000

For cities that are far away:

1-Dimension: GOF = 0.6126508 0.6535741
 2-Dimension: GOF = 0.8809303 0.9397740
 3-Dimension: GOF = 0.9373853 1.0000000

So, from both graph and data, I think that 3-dimensional model fit the data better than other two, and 2-dimensional model is clear to see. In my opinion, I think it is because I choose places as my data, and the earth looks like a 3d model, so 3-dimensional model fit them well. Also, a map is in 2d, so the 2-dimensional model is easy for us to see.