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=",")
model2 <- cmdscale(distances,k=2)
plot(model2,asp=1,
     ylab="",xlab="")
```

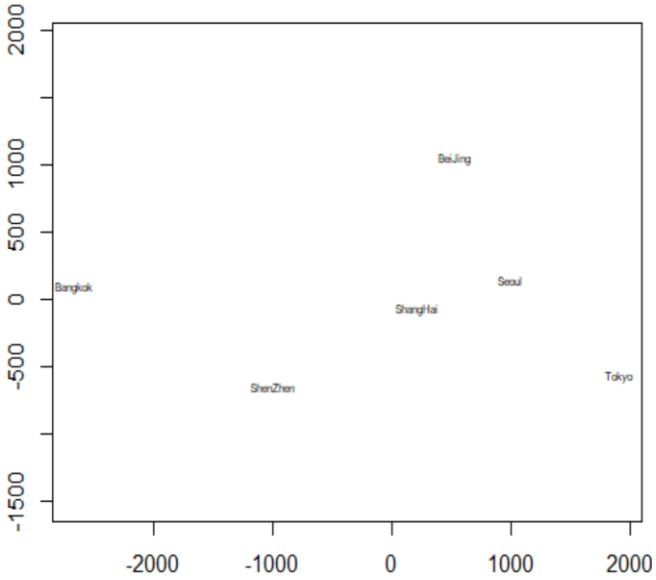
Also I use following code to generate the text plot which is clear to see the city position:

```
distplusnames <- read.csv(file="381name.csv",head=TRUE,sep=",")
text(model2[,1],model2[,2],
```

cex=1)

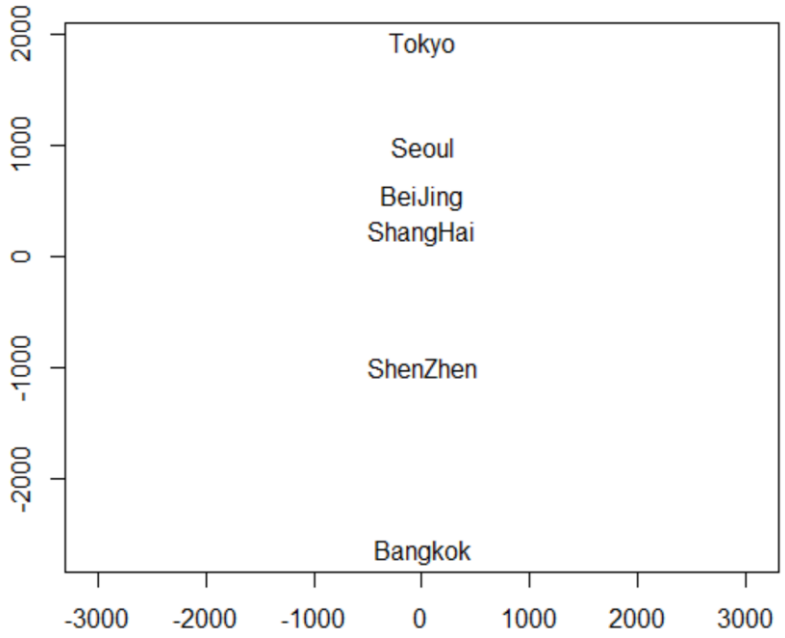
This creates the following textplot with name in each position:

They create the plot below:

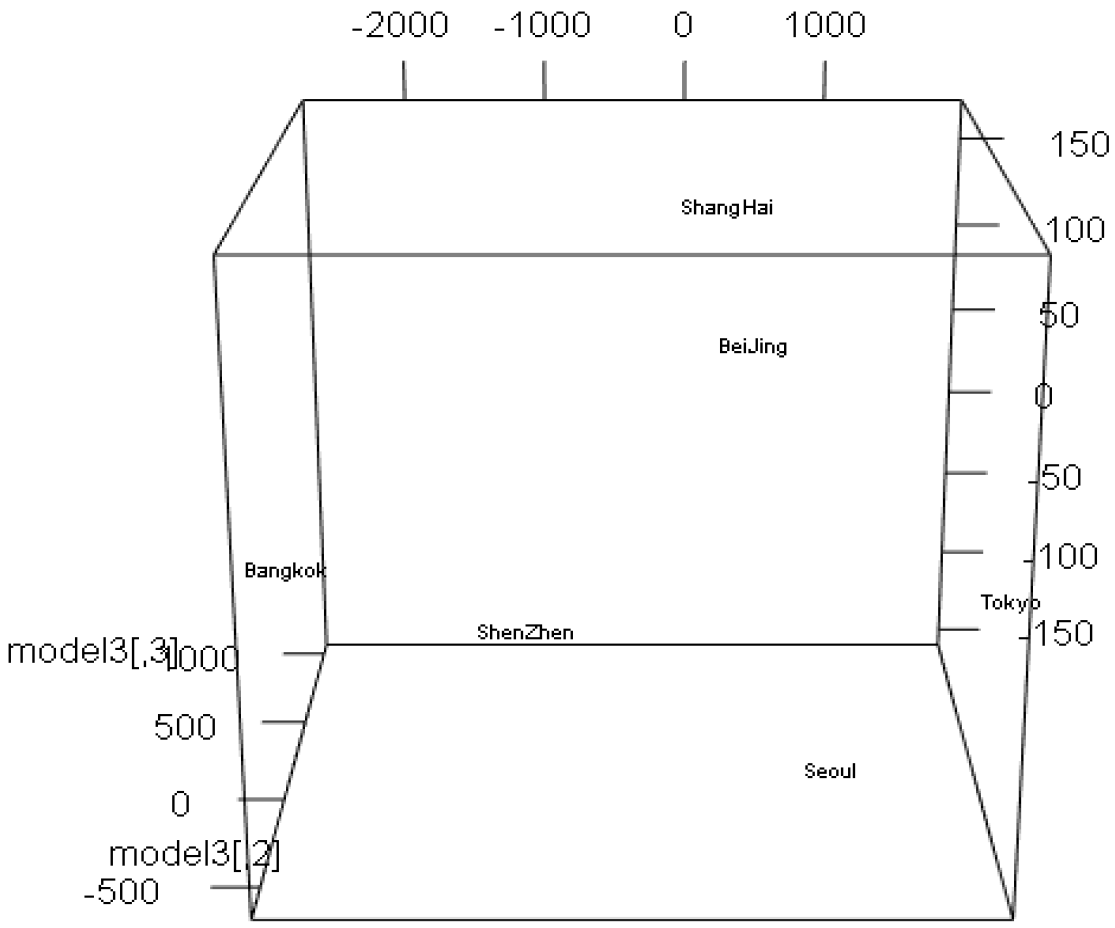


The city that far away from other is Bangkok, and others seem to be close.

Then I change the k of the code in the “cmdscale” part to 1 and 3 to creates 1 and 3-dimensional scale, it creates the following results (in a line):

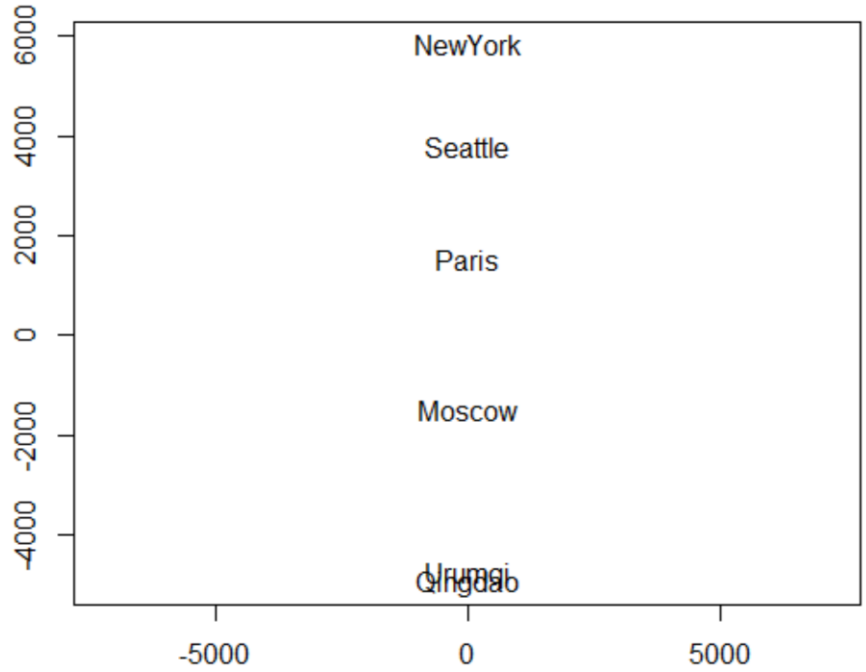


It is seems that BeiJing, ShangHai, and Seoul are pretty close to each other.

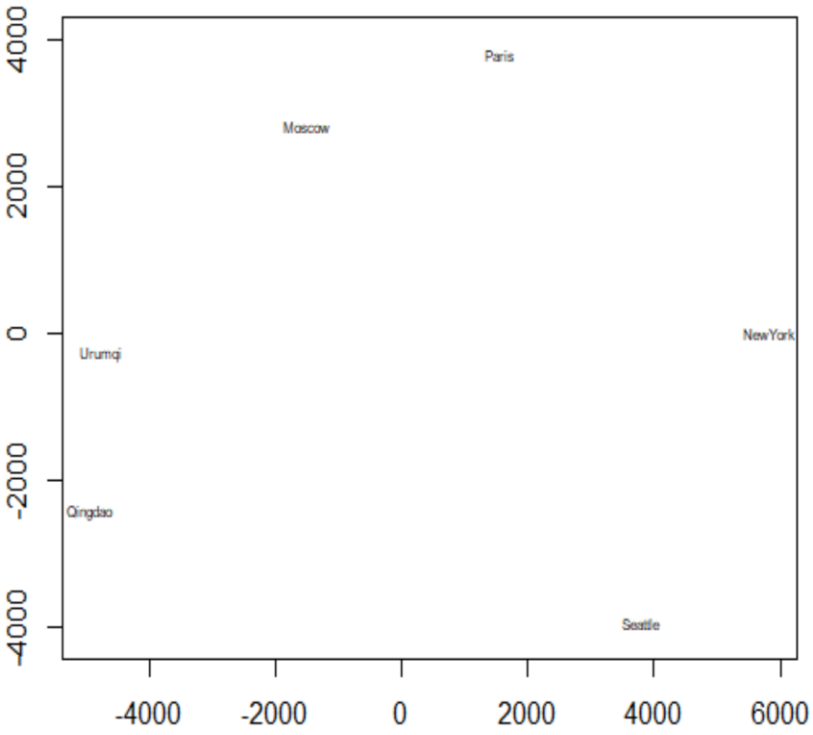


It is not so clear, to see, but I see that Bangkok and Tokyo has greatest distance

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

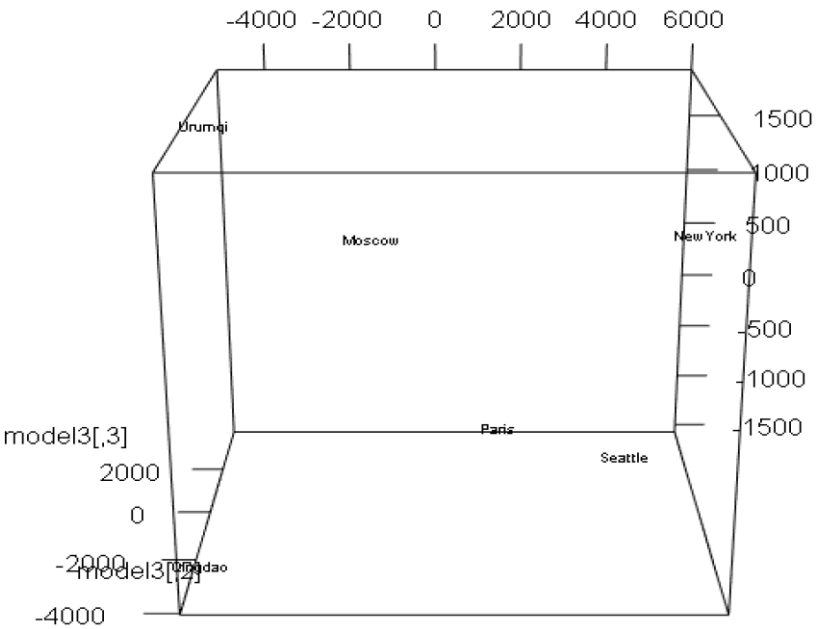


I see that Urumqi and Qingdao are close
The plot with text in 2-dimensional:



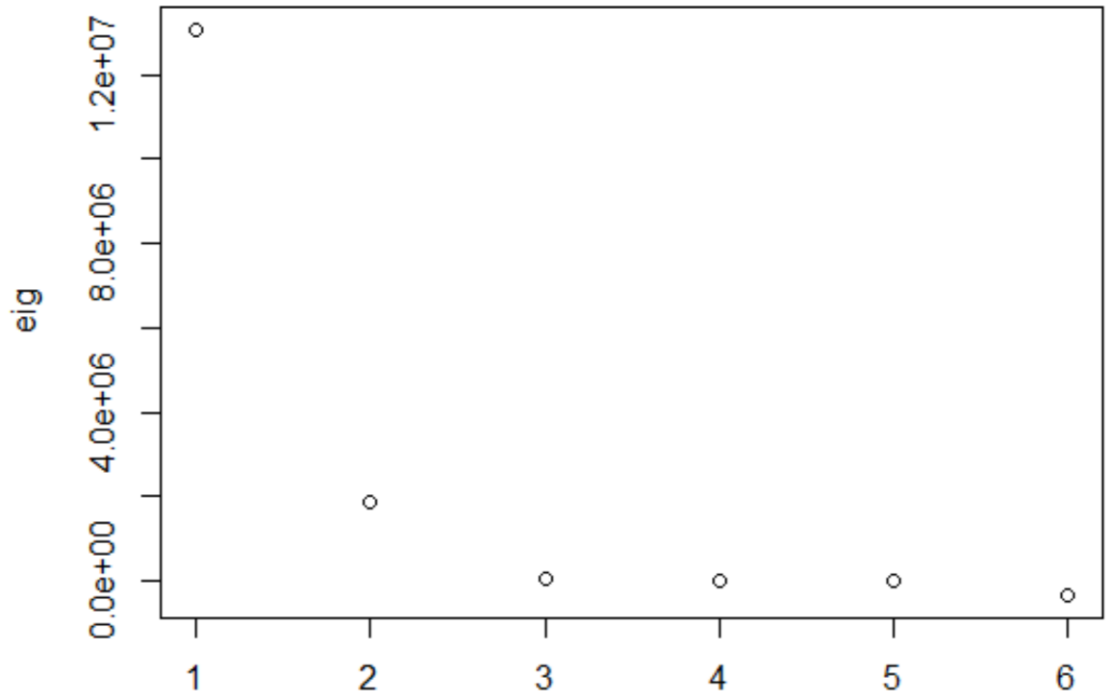
I see that Seattle and New York are far away from other, and I think this is because they are cities from America

The plot with text in 3-dimensional:

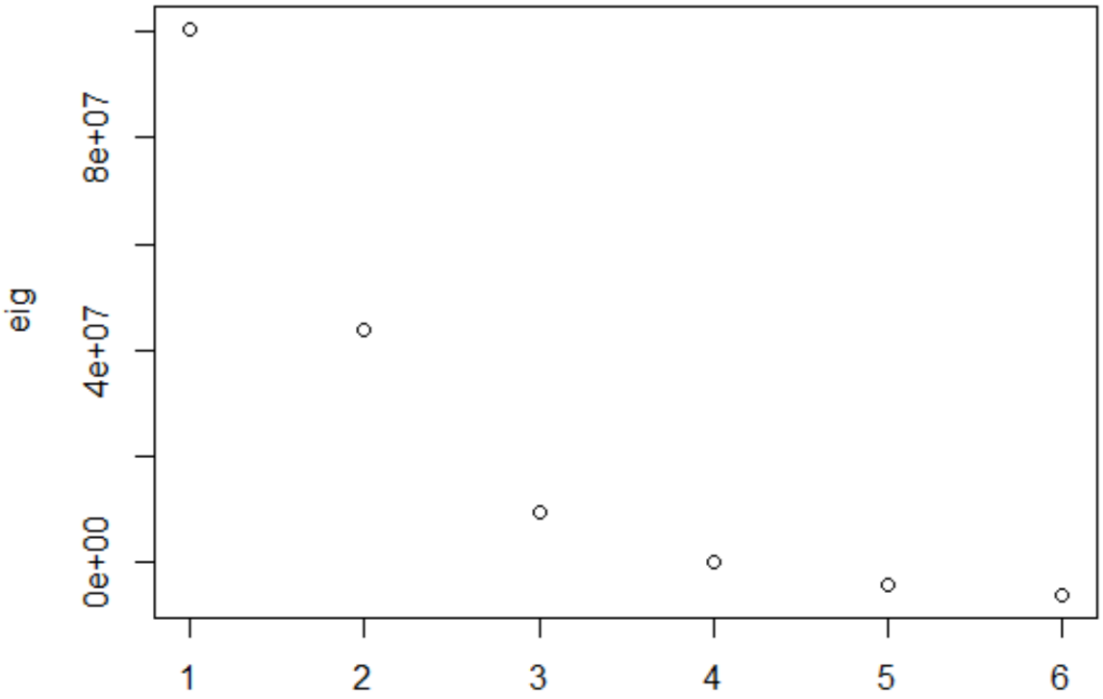


I see that New York and Seattle are still close.
To see how good the models are, I will investigate the eigenvalue for them from the

largest to smallest:
Eigenvalue for cities that are close together:

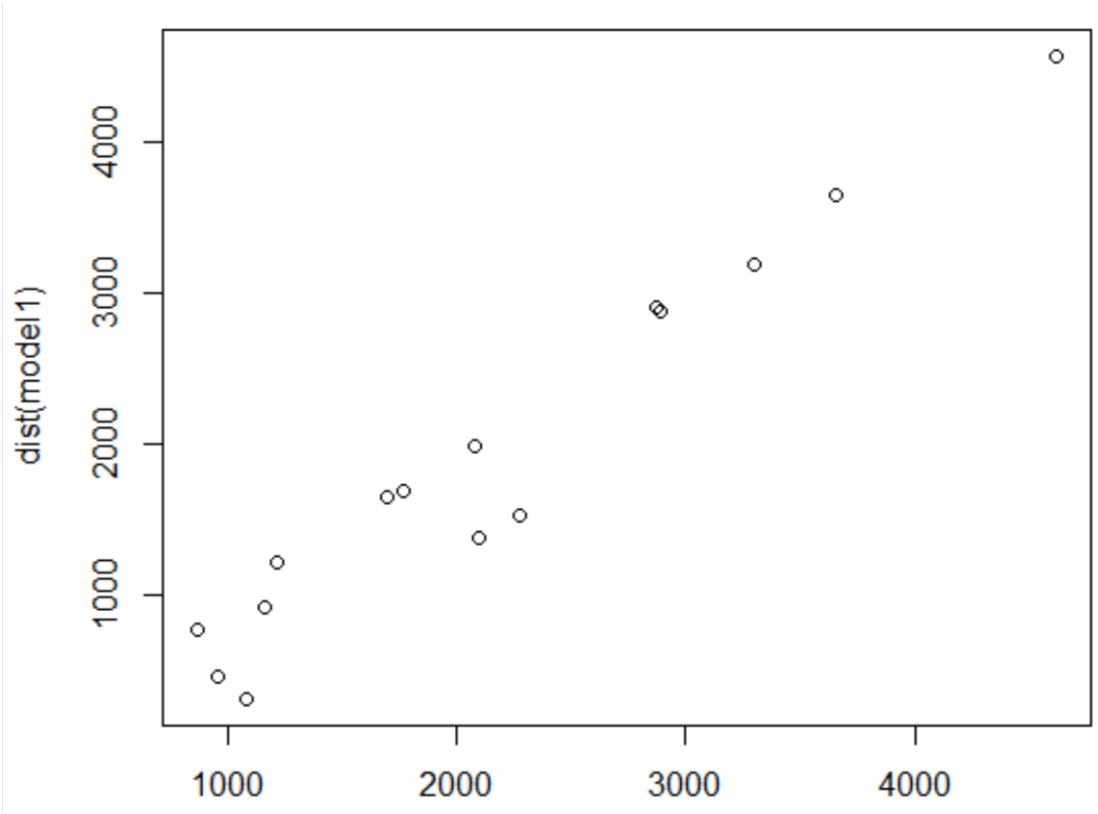


In the plot, I see that the first eigen value seems to be larger than other, and the third to sixth seem to be close to 0
Eigenvalue for cities that are far away:

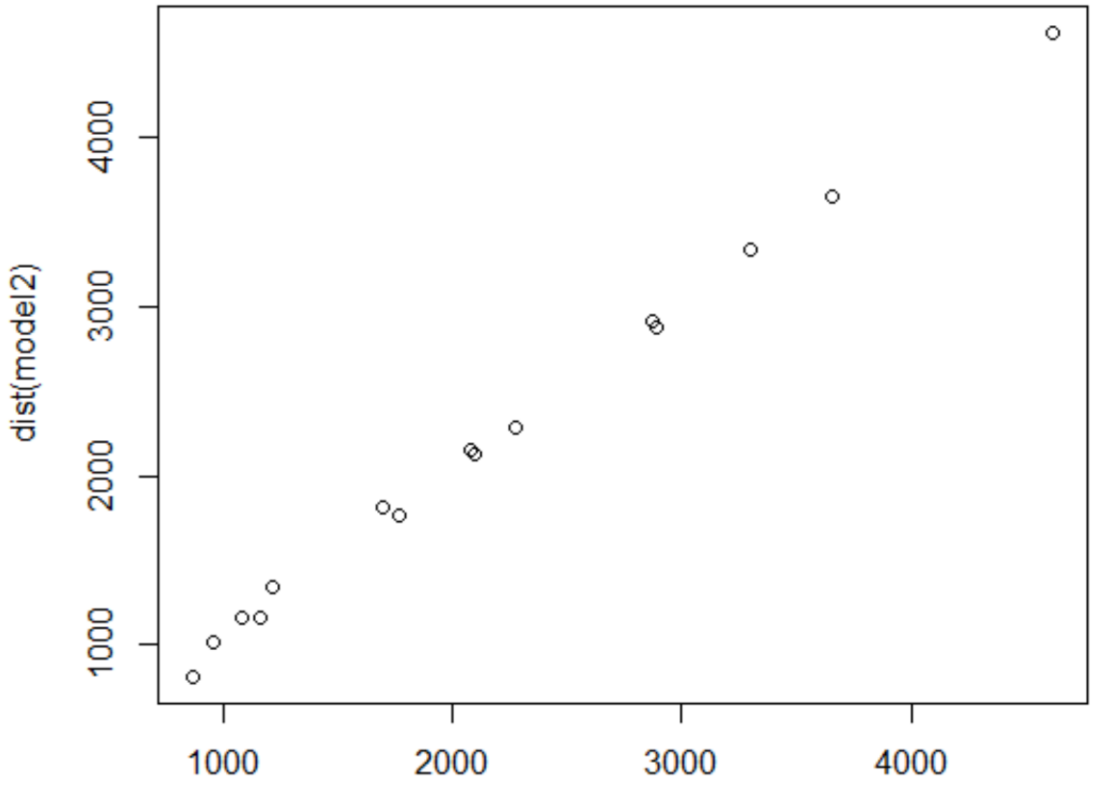


In the plot, the last two eigenvalues seem to be zero
We see that about the first 3 are clearly non-zero, which means that 20-dimensional space fits the data best. Also, because first two eigenvalues are not small, 1-dimensional model and 2-dimensional model will not fit them perfectly.

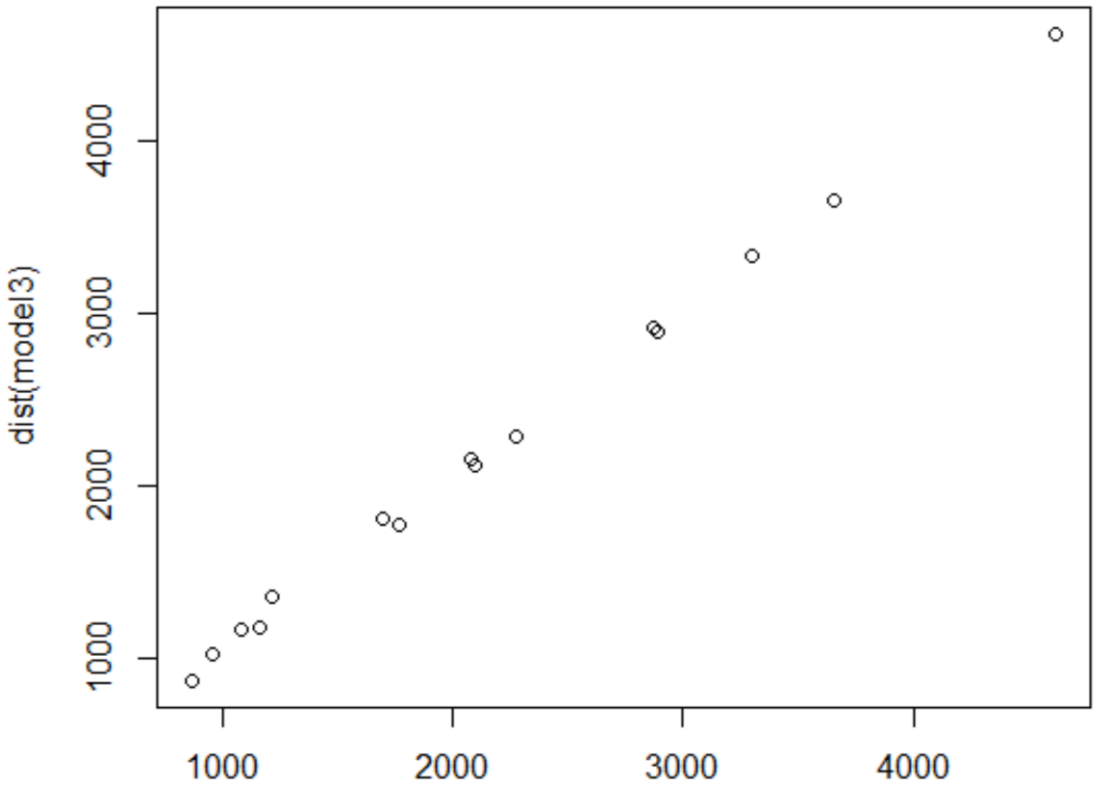
Also, there is another way to see how good each model is, which is comparing the distance in model with original distance in plot. The plot is shown below:
Original distance vs distance in 1,2,3-dimendional model of close together cities:



I see that the original number less that 3000 is disarray

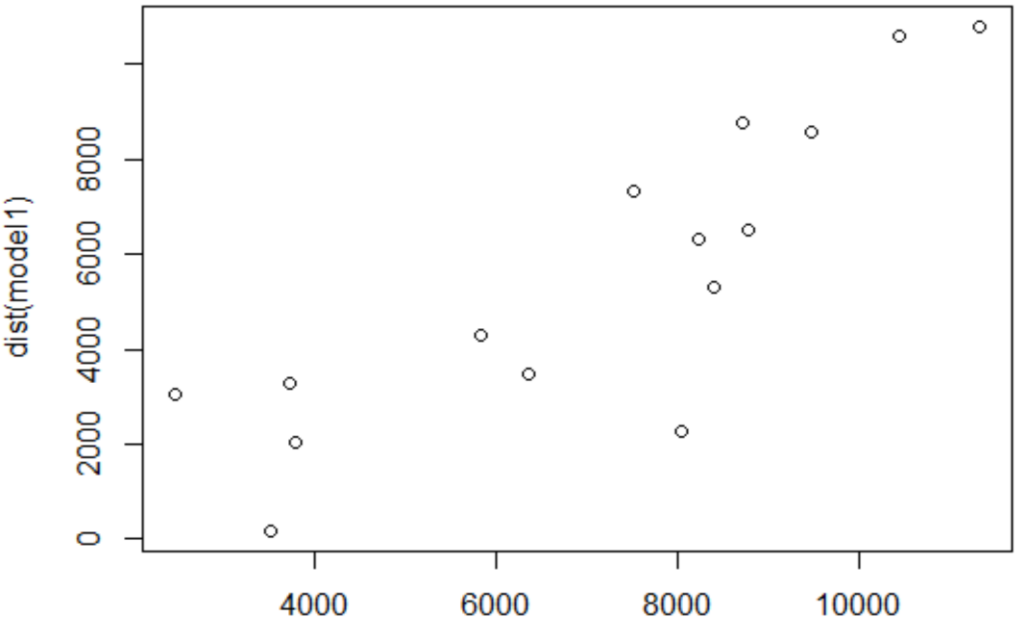


I see that all points looks like in a line

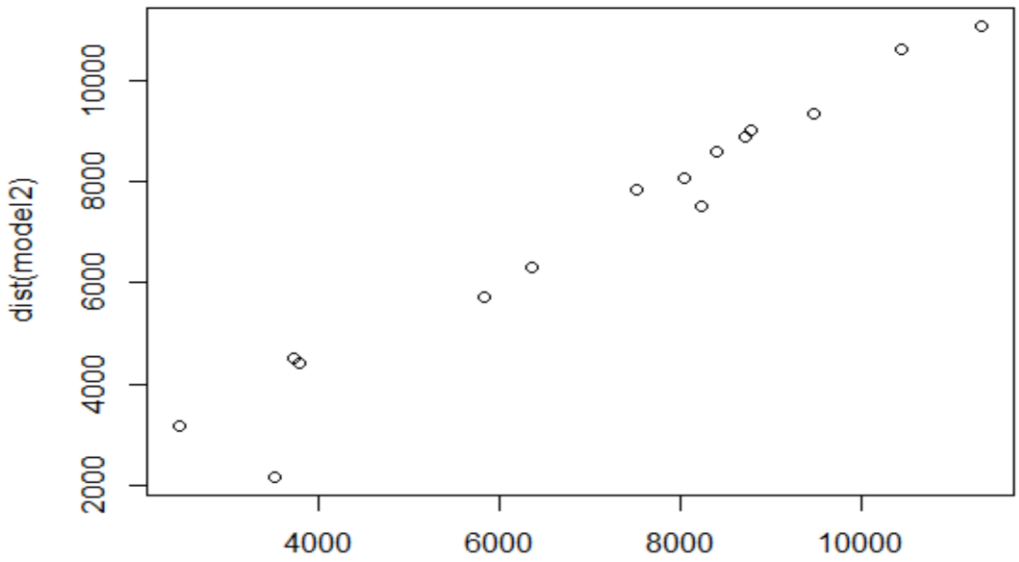


This plot is like the model2

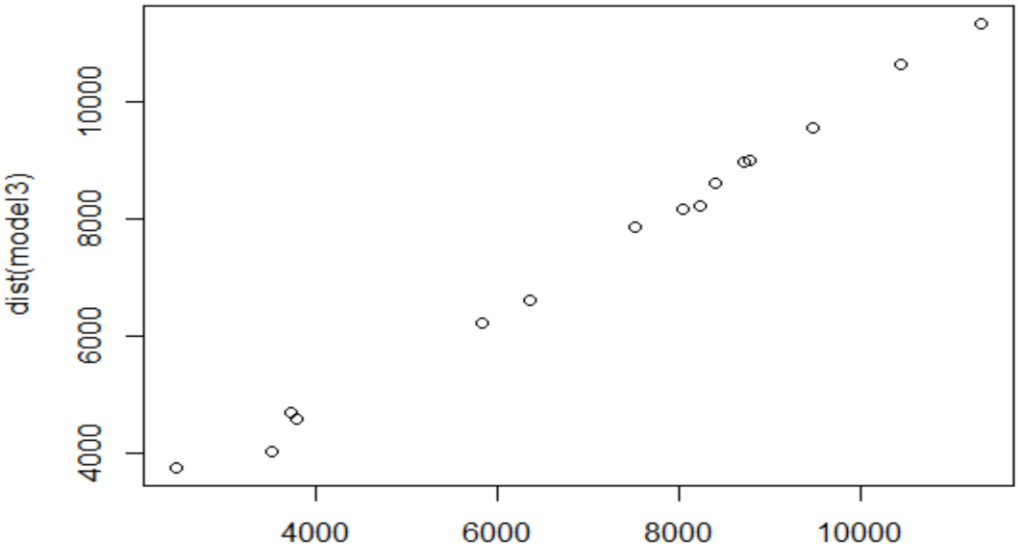
Original distance vs distance in model among cities that are far away



I see that all the points are scattering



I see that some points are scattering



I see that all points almost lie in a line.

Also, there is a table comparing each of their model3 with original data:

Model3 for cities that are close together:

```
          [,1]       [,2]        [,3]
[1,]   533.4825 1045.87500   31.2145254
[2,] -1000.7534 -644.57126   -8.5490170
[3,]   216.9968  -72.54574  167.0482477
[4,] -2660.0559   92.53882  -34.7674900
[5,]   995.2061  145.96600 -155.2323929
[6,]  1915.1239 -567.26281    0.2861268
```

Original distances for cities that are close together:

```
     BJ     SZ    SH   BK   SE    TO
1   0.0 2275.6 1084 3301  954 2100
2 2275.6    0.0 1215 1698 2082 2873
3 1084.0 1215.0    0 2886  868 1765
4 3301.0 1698.0 2886    0 3654 4620
5  954.0 2082.0  868 3654    0 1160
6 2100.0 2873.0 1765 4620 1160    0
```

Model3 for cities that are far away:

```
         [,1]         [,2]      [,3]
[1,]  1549.424  3.806092e+03 -1503.8071
[2,] -4949.443 -2.434408e+03 -1527.5071
[3,] -4763.730 -2.607666e+02  1862.4086
[4,]  5841.447 -2.353875e-02   926.9819
[5,] -1485.083  2.826413e+03   509.0773
[6,]  3807.385 -3.937307e+03  -267.1536
```

Original distances for cities that are far away:

```
   Pa    QD     UR    NY    MO  SEA
1   0  8780  8229  5840 2477 8044
2 8780     0  3519 11321 6352 8702
3 8229  3519     0 10441 3736 9457
4 5840 11321 10441     0 7511 3800
5 2477  6352  3736  7511    0 8404
6 8044  8702  9457  3800 8404    0
```

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 model3 vs original distance and model2 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.

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, so the 3-dimensional model is exactly like the earth, while 2-dimensional model is like a map.