HACETTEPE UNIVERSITY

GEOMATICS ENGINEERING

GMT 352 – GEOGRAPHICAL INFORMATION SYSTEMS

Moran I Project

Name: Berk

Surname : KIVILCIM

Number: 21632734

My own Implementation (In matlab code there is no library or plugins needed so It will run directly)

Step – 1 Adding datas to postgres: I did this because I want to copy paste datas easly. Firstly I created a csv with attribute table of p_iller and then imported it to postgres.

Column 6 for 2007-2008 density data and column 7 for 2017 density data.

1	1	Çanakkale	17	Tekirdağ,Edirne,Balıkesir	-2.812018439049682	53.3994764924998
2	2	Çankırı	18	Karabük,Kastamonu,B...	11.88800221637939	24.8429906542056
3	3	Çorum	19	Amasya,Kastamonu,Sa...	-8.005360152821742	41.3087867417136
4	4	Adana	1	Kahramanmaraş,Kayse...	9.820961764359561	159.286740927057
5	5	Adıyaman	2	Diyarbakır,Kahramanm...	3.947500863890668	87.455708801365
6	6	Afyonkarahisar	3	Konya,Kütahya,Eskişeh...	-6.014584906147468	49.9995109682828
7	7	Ağrı	4	Iğdır,Erzurum,Kars,Bitli...	2.447654644804524	46.7554489973845
8	8	Aksaray	68	Konya,Nevşehir,Niğde,...	12.18681391145471	53.1577278731836
9	9	Amasya	5	Tokat,Samsun,Çorum,Y...	-16.28134999132865	57.9768014059754
10	10	Ankara	6	Aksaray,Konya,Bolu,Ça...	18.23160086864141	222.055625790139
11	11	Antalya	7	Mersin,Karaman,Konya...	38.36493927006737	114.095256478309
12	12	Ardahan	75	Erzurum,Kars,Artvin	-4.258484496349777	20.0528707145808
13	13	Artvin	8	Erzurum,Ardahan,Rize	-9.011761856893763	22.5523279489616

Step – 2 Connecting with python: I connect database with python and got datas in a list (matrices) format. Results in Python Shell and I copied them. Just used pyscopg2 library.

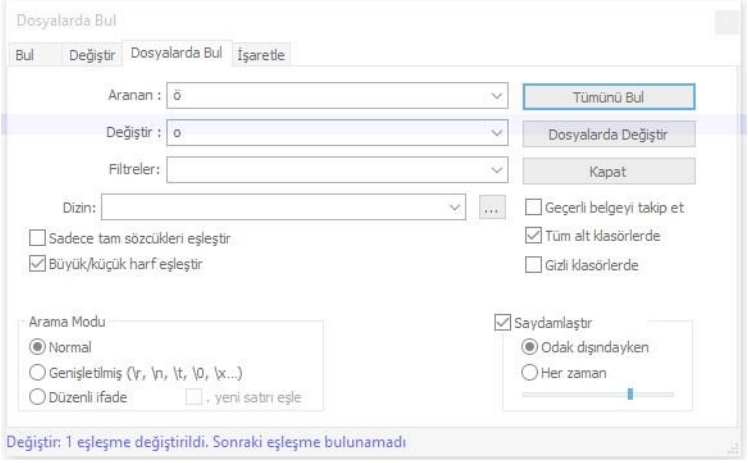
An example of 2017 density list.

```
[53.3994764924998, 24.8429906542056, 41.3087867417136, 159.286740927057, 87.4557
08801365, 49.9995109682828, 46.7554489973845, 53.1577278731836, 57.9768014059754
, 222.055625790139, 114.095256478309, 20.0528707145808, 22.5523279489616, 137.66
8959368233, 84.2593188334849, 93.0658653846154, 125.617514488088, 21.50762235891
95, 51.5325430032543, 33.1217739003999, 48.6360917248255, 36.4403846153846, 38.7
103801169591, 281.78881212819, 147.101675107129, 87.1309442353746, 112.890224465
4, 66.9830424761278, 69.0326434062685, 19.9252087098718, 30.0310389764246, 62.17
45412512643, 26.4366941121641, 294.106907171139, 64.0212236533958, 38.4121743975
484, 270.285861358957, 54.2851170568562, 52.4202513291445, 2892.4617013087, 356.
283466533467, 78.601909940053, 47.7795775235869, 59.4920905329764, 27.8882984737
14, 28.4046608077417, 28.3108796472288, 80.7793228891627, 95.4614845938375, 61.4
797088663432, 56.7139216310927, 36.9220717884131, 521.392580287929, 56.083888560
1832, 66.8033288043478, 107.898671350031, 91.9508289802407, 115.849596383597, 73
.0488677923897, 50.1977912892418, 54.3530395984384, 47.9770130576714, 124.721270
16129, 168.925736235595, 84.4061703212647, 204.674245556015, 144.554662556424, 1
05.822168931521, 59.2716974237164, 35.8126726519337, 70.3629753914989, 21.762618
6556447, 159.268651987961, 60.4625426792529, 168.594768439108, 11.1003767491927,
 68.3338326156151, 57.3548370381885, 296.579693034239, 29.7505685048323, 180.657
384987893]
PostgreSQL connection is closed
```

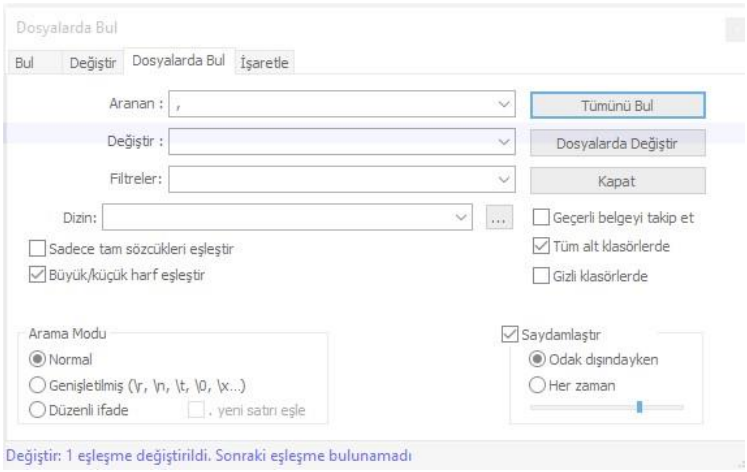
Step – 3 Converting the lists (Matrices) in matlab matrix format: Because my Moran code work on matlab. I used notepad++ for this step. Ara>değiştir.

Example:

(Making all “ö”’ to “o”)



(Making all “,“ to “space“)



Step – 4 Making Weight Matrix:

Matching city names and id’s. This matching made by human hand.

```
Adana=01;
Adiyaman=02;
Afyonkarahisar=03;
Agri=04;
Amasya=05;
Ankara=06;
Antalya=07;
Artvin=08;
Aydin=09;
Balikesir=10;
```

1-I identfy Cities id (plaka) numbers because mainly I work with this values.

```
plaka=[17 18 19 1 2 3 4 68 5 6 7 75 8 9 10 74 72 69 11
```

…

2-Copy pasting neighbor, plaka and density matrices. I manuelly added zeros to neighbor matrix and making it 81x9 so generating this matrix called semi automatic. Its 81x9 because maximum neigborhood is 9.

```
komsu2=[Tekirdag Edirne Balikesir 0 0 0 0 0 0 ;
    Karabuk Kastamonu Bolu Ankara Kirikkale corum 0 0 0;
    Amasya Kastamonu Samsun Sinop cankiri Kirikkale Yozgat 0 0;
    Kahramanmaras Kayseri Hatay Osmaniye Mersin Nigde 0 0 0;
    Diyarbakir Kahramanmaras Malatya Gaziantep sanliurfa 0 0 0 0;
    Konya Kutahya Eskisehir Isparta Denizli Burdur Usak 0 0;
    Igdir Erzurum Kars Bitlis Mus Van 0 0 0;
    Konya Nevsehir Nigde Ankara Kirsehir 0 0 0 0;
    Tokat Samsun corum Yozgat 0 0 0 0 0;
    Aksaray Konya Bolu cankiri Kirikkale Kirsehir Eskisehir 0 0;
    Mersin Karaman Konya Isparta Burdur Mugla 0 0 0 ;
    Erzurum Kars Artvin 0 0 0 0 0 0;
    Erzurum Ardahan Rize 0 0 0 0 0 0;
    Denizli Mugla Manisa Izmir 0 0 0 0 0;
    Kutahya Bursa Manisa Izmir canakkale 0 0 0 0;
    Karabuk Kastamonu Zonguldak 0 0 0 0 0 0;
    Mardin Bitlis Mus Siirt Diyarbakir 0 0 0 0;
    Erzurum Rize Erzincan Gumushane Trabzon 0 0 0 0;
    Sakarya Bolu Kutahya Eskisehir Bursa 0 0 0 0;
    Erzurum Mus Elazig Erzincan Diyarbakir Tunceli 0 0 0;
    Agri Batman Mus Siirt Van 0 0 0 0;
```

In the end neighbor matrix means like this with integer values.

```
komsu2=[59 22 10 0 0 0 0 0 0;      %çanakkale
    78 37 14 06 71 19 0 0 0;        %çankırı
    05 37 55 57 18 71 66 0 0;       %çorum
    46 38 31 80 33 51 0 0 0;        %adana
    21 46 44 27 63 0 0 0 0;         %adıyaman
    42 43 26 32 20 15 64 0 0;       %afyonkarahisa
    76 25 36 13 49 65 0 0 0;        %ağrı
    42 50 51 06 40 0 0 0 0;         %aksaray
    60 55 19 66 0 0 0 0 0;          %Amasya
    68 42 14 18 71 40 26 0 0;       %Ankara
    33 70 42 32 15 48 0 0 0;        %Antalya
    25 36 08 0 0 0 0 0 0;           %Ardahan
    25 75 53 0 0 0 0 0 0;           %Artvin
    20 48 45 35 0 0 0 0 0;           %Aydın %%%
    43 16 45 35 17 0 0 0 0;         %Balıkesir
    78 37 67 0 0 0 0 0 0;          %Bartın
    47 13 49 56 21 0 0 0 0;         %Batman
    25 53 24 29 61 0 0 0 0;         %Bayburt
    54 14 43 26 16 0 0 0 0;         %Bilecik
    25 49 23 24 21 62 0 0 0;        %Bingöl
    04 72 49 56 65 0 0 0 0;            %Bitlis
```

Code's first step (making a matrix that shows how many neighborhoods city has):

Workflow:

In this part we search neighbor matrix (komsu2) and if the element of the neighbor matrix not zero that means there is a neighborhood it adds +1 to 'komşuluk' data that shows how many neighborhood the city has and adding this values to a new matrix. Then the code jump to second row and komşuluk value is reset to 0. When code jumps to another row it saving the values in another column of 'yeni'. In the end there is a matrix that show how many neighbor the city has. We will use this data on weight matrix. Matrix looks like below here.

```
Command Window

  Columns 1 through 22

     3     6     7     6     5     7     6     5     4     7     6     3     3     4     5     3     5     5     5     6     5     8

  Columns 23 through 44

     5     6     3     6     8     3     5     9     9     6     4     6     5     2     3     2     4     2     3     7     7     5

  Columns 45 through 66

     3     4     5     6     1     5     2     5     4     9     6     6     5     5     4     6     5     6     4     4     4     5

  Columns 67 through 81
```

```
for i=1:r2
    komsuluk=0;
    for j=1:c2
        if komsu2(i,j)~=0
            komsuluk=komsuluk+1;
            yeni(1,i)=komsuluk;
        end
    end
end
```

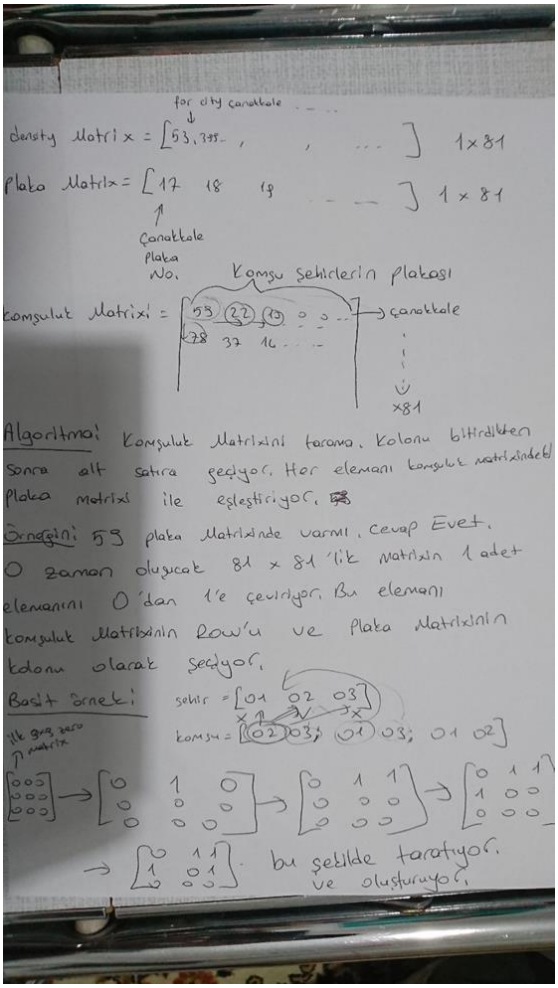
Code's second step (generating weight matrices which city has which neighbor):

Workflow:

Firstly created a 81x81 zeros matrix. After that checking the neightbor matrix (komsu2) and when we are on the element we search the plaka matrix. If plaka matrix value and komsu2 value is the same we take the komsu2's row number and plaka's column number and make element of the 81x81 zeros matrix 1 for this row and column. We take Komsu2's row number because each city has its own row in 81x81 matrix. Then 81x81 zeros matrix looks like diagonal matrix with neihgbor values 1. It looks confused so I shows basicly in figüre with simple matrices below here.

```
for i=1:r2
    for j=1:c2
        for k=1:cc2
            if komsu2(i,j)==plaka(1,k)& plaka(1,k)~=0
                bos2(i,k)=1;

            end
        end
    end
end
```



Merging the first and second step:

Our weight matrices 1 values must be (1/(how many neighbor the city has)) so we need to merge our komsuluk and komsu2 matrix.

Workflow: We are dividing all values in the row of the 81x81 matrix (bos2) with how many 1 values in this rows. Matrix named komşuluk (1x81). Actually 1x81 matrices columns means cities and 81x81 matrices rows means cities and columns means neighborhood so in code i's location is different.

```
for i=1:81
    for j=1:81
        bos2(i,j)=bos2(i,j)/yeni(1,i);
    end
end
```

Step - 5 Moran's Formula:

Part – 1 Calculating the total values of the weight matrix: Searching all values in the weight matrix and added them to together.

```
for i=1:81
    for j=1:81
        W=W+bos2(i,j);
    end
end
```

Part – 2 Calculating mean value of the density matrix: Added all values in the density matrix and dividing with the lenght of the matrix.

```
for i=1:ccc
    A=A+density(1,i);
end
ortalama=A/ccc;
```

Part – 3 Calculating the last result:

The Upper part of the Moran code below here:

```
for i=1:81
    for j=1:81
        ust=ust+(bos2(i,j)*((density(1,i)-ortalama)*(density(1,j)-ortalama)));
    end
end
```

The lower part of the Moran code below here:

```
for i=1:81
    alt=alt+((density(1,i)-ortalama)^2);
end
```

The Last Result for 2017 data:

```
disp(((ust)/(alt))*N/W)

0.147039828952744
```

The Results with Pysal for 2017 data:

```
0.14703982895274478
```

With Different Data: The other density matrix in saved matlab code as comment.

I also use 2007-2008 density data.

Matlab (My own implementation) Result:

0.242862294874132

Python (with pysal) Result:

0.24286229487413238