# Steps for creating and using eigenvectors for space time analysis


## Author: Li An


## Date: Nov. 7, 2014

**Input data: climate_change_wCounty.shp under D:/2.Papers/2.CDI-ClimateChange/Paper-1-Methodology/Revision/ -- created in the CDI project by students;

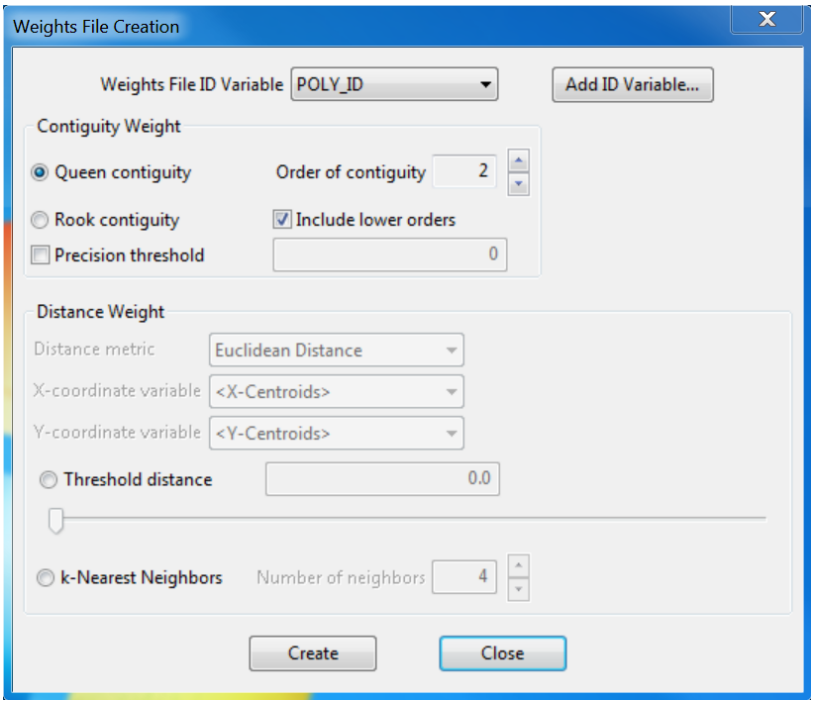
**Step 1: Prepare the data in shapefile.**

Go to ArcGIS, add: climate_change_wCounty.shp that is saved under D:\2.Papers\2.CDI-ClimateChange\Paper-1-Methodology\Revision\High_order_test\. Open its attribute table and review the data. Close ArcGIS.

Note: throughout this whole procedure, we will use RowID to link all the data. This is very important!!

**Step 2: Prepare the GAL files.**

This step facilitates creating spatial matrices in a later step. Open GeoDa/File/New Project, choose ESRI shapefile (*.shp), then browse to the directory D:\2.Papers\2.CDI-ClimateChange\Paper-1-Methodology\Revision\High_order_test\ and choose the above-mentioned shapefile: climate_change_wCounty.shp.




Next go to Tools/Create, choose Add ID Variable, and choose POLY_ID (which is equal to RowID) as ID variable (as shown below). Then choose Queen continuity and click Create.

When prompted for file name, choose Queen_conn1.GAL or whatever name you choose. Note when creating weights files with order of contiguity higher than 1, make sure to check the box next to Include lower orders (as shown below).

Here I created a bunch of files Queen_conn1.GAL, Queen_conn2.GAL, … Queen_conn20.GAL, which will be used later.

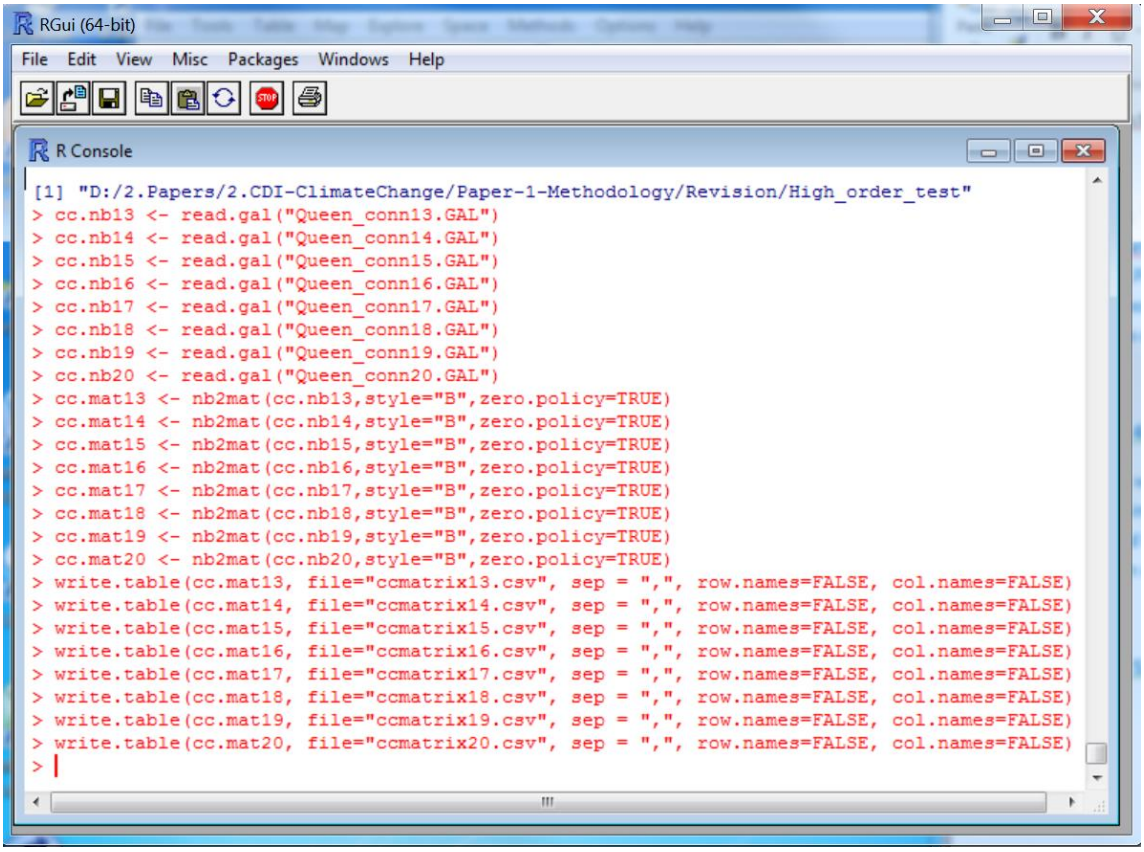
**Step 3. Create and output spatial matrices in R.**

Next open R console and run the following commands:

```
setwd("D:/2.Papers/2.CDI-ClimateChange/Paper-1-Methodology/Revision")

getwd()

library(spdep)

library(RColorBrewer)

library(classInt)

library(maptools)

gpclibPermit()

cc <- readShapePoly("climate_change_wCounty.shp")

cc.nb1 <- read.gal("Queen_conn1.GAL")

cc.mat1 <- nb2mat(cc.nb1,style="B",zero.policy=TRUE)

write.table(cc.mat1, file="ccmatrix1.csv", sep = ",", row.names=FALSE, col.names=FALSE)
```

Note: The above yellow option has to be there. If not, there is an error message and the matrix Named cc.mat will not be created. cc.nb1 and cc.mat1 are R objects created and saved in the working directory D:/2.Papers/2.CDI-ClimateChange/Paper-1-Methodology/Revision /High_order_test/. The numbers represent the order of connectivity. The file ccmatrix1.csv is the output csv file that is also saved in the same directory.

Below is what the R window looks like.




### Step 4. Modify the CSV files for next steps.

Next we need to add a row of v1, v2, v3, …, v3109 as the top row within the file ccmatrix.csv before running the SAS code named LTM_Eigen_Reg.sas. When saving the changes to ccmatrix.csv, you will be asked whether you want to keep the workbook in this format. Simply click Yes. Then close the file—when asked again about whether to save the changes, click “No”.

### Step 5. Implement eigenvector filtering in SAS

Below are instructions about how to run the SAS code:

5.1 Import the data.

Open the SAS code, and change the directory at import procedure to where the datafile ccmatrix.csv is saved. Run Step 1.

```
* Step 1: import the binary (0/1) matrix generated in R;
proc import datafile="D:\2.Papers\2.CDI-ClimateChange\Paper-1-
Methodology\Revision\High_order_test\ccmatrix2.csv"
      out= conn replace;
run;
```

## 5.2 Generate eigenvectors.

Run Step 2—this may take a few minutes. Record the Sum_C value from the Output panel and open the Eig_value file to record the biggest eigenvalue (the one on the top). Put these two numbers in Table 1 below.

Note: if we want to test the effects of neighborhood definition (from 1st order to 12th order Queen's definition), replace the csv file with the one we want to use (highlighted in yellow below). Also if too many such changes are needed, we can write a SAS Macro to automate this process. Now it is not too demanding, so we leave it as is.

## 5.3 Calculate the Moran's I and adjusted eigenvalue:

Run Step 3. Put the Sum_C and maximum eigenvalue in the yellow places below (Step 3) and run the data procedure below.

```
data Moran_I;
      set Eig_value;
      MI = eigen*3109/1512682; *18174 for the 1st order nb, 38844 for
the 2nd order nb, 82174 for the 4th order nb,
      122894 for the 6th order, and 141914 for the 7th and 8th order;
      adj_eigen=eigen/499.2477;
run;
```

Table 1. Relationships between order, maximum eigenvalue, and # of eigenvectors with Moran's I greater than 2.5 and adjusted eigenvalue greater than 2.5.

| Order # | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Sum_C | 18174 | 57018 | 117494 | 199668 | 302740 | 425634 | 567548 |
| Max. Eigen | 6.6516 | 20.9021 | 43.4480 | 74.0030 | 111.5159 | 154.9512 | 203.7122 |
| # of records with I >0.25 | 793 | 310 | 158 | 93 | 61 | 44 | 32 |
| # of records with adjusted eigen > 0.25 | 744 | 288 | 145 | 85 | 56 | 40 | 30 |

| Order # | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|
| Sum_C | 727272 | 902854 | 1093296 | 1297070 | 1512682 | 1739432 | 1976712 |
| Max. Eigen | 257.2880 | 314.5046 | 374.6171 | 436.5997 | 499.2477 | 561.7657 | 623.3854 |
| # of records with I >0.25 | 25 | 19 | 16 | 13 | 10 | 9 | 8 |
| # of records with adjusted eigen > 0.25 | 23 | 18 | 15 | 13 | 10 | 9 | 8 |

| Order # | 15 | 16 | 17 | 18 | 19 | 20 | |
|---|---|---|---|---|---|---|---|
| Sum_C | 2222940 | 2477150 | 2738030 | 3004064 | 3273966 | 3546650 | |
| Max. Eigen | 683.1172 | 740.2435 | 794.0626 | 844.0188 | 889.5929 | 930.3354 | |
| # of records with I >0.25 | 7 | 6 | 5 | 4 | 4 | 4 | |
| # of records with adjusted eigen > 0.25 | 7 | 7 (no change) | 6 | 5 | 4 | 4 | |

### 5.4 Run Steps 4 through 10.4 to calculate residuals.

Due to the variable RepubPercEvan has dots (.) as some counties have no data, comment out the terms under proc mixed that contains this variable under Step 8.2: An example is below (the highlighted lines):

```
* 8.2--M2:  Intercept, time, time square, and covariates--with eigen vectors--this is new M2 (former M4);
proc mixed data = sampleCntydataLTM noclprint covtest method=ml;
      class CNTYIDFP;
      model edu = Stage Stagesq /*RepubPercEvan*/ Pop_density POP_urban SummerDays MaxDrySpel N10_16over medhhldinc
       /*Stage*RepubPercEvan*/ Stage*Pop_density Stage*POP_urban Stage*SummerDays Stage*MaxDrySpel Stage*N10_16over
            Stage*medhhldinc col1--col100
      /*Stagesq*RepubPercEvan*/ Stagesq*Pop_density Stagesq*POP_urban Stagesq*SummerDays Stagesq*MaxDrySpel
            Stagesq*N10_16over Stagesq*medhhldinc /solution outp=test ddfm=bw notest; * I added outp=test to report the predicted values and residuals Nov. 8, 2014;
      random intercept Stage Stagesq/sub=CNTYIDFP type=cs g;
      repeated/sub=CNTYIDFP type = cs  r rcorr;
     title "M2: Intercept, slope, slope-square, eigenvectors, and covariates: random coefficient model";
run;
```

When these changes are made, run Steps 4 through 10.4 at tone time (highlight all of these sections and hit the run botton).

5.5 Go to Windows Explorer or Computer under the data storage directory (here D:/2.Papers/2.CDI-ClimateChange/Paper-1-Methodology/Revision /High_order_test), change the output dbf file to a different name, say, from eData.dbf to eData1.dbf.

Repeat steps 5.1-5.3 for the 2nd order connectivity. Fill in Table 1, and then run Steps 4 to end and obtain residuals. Repeat the above steps until all csv files (for various orders of connectivity) are used as input files.

Note: eData1.dbf, eData1.dbf, etc. will be used in ArcGIS to attach the residuals to the records or counties. This change of file names guarantees that the file will not be overwritten next time you run the SAS code with a different neighborhood as input file.

Below we will do some operations in ArcGIS. The purpose is to link all the residuals under the 1st order, 2nd order, …, 12th order definitions to the shapefile named climate_change_wCounty.shp.

**Step 5. Add residual data to the shapefile.**

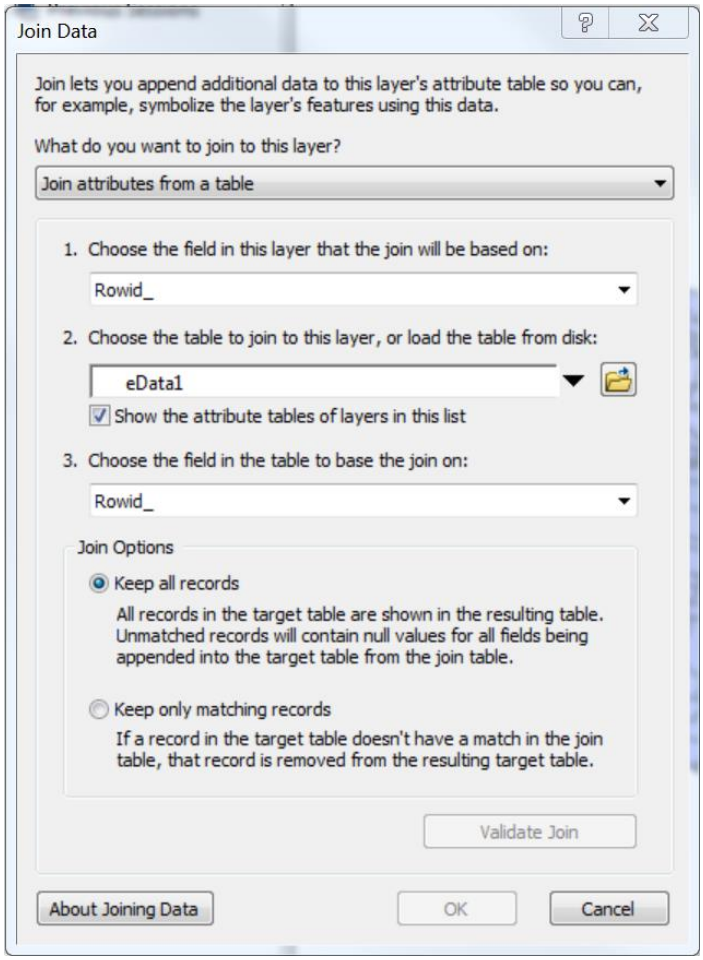
First open ArcGIS and put in the shapefile climate_change_wCounty.shp. Too make it less messy, for this shapefile saved in D:\2.Papers\2.CDI-ClimateChange\Paper-1-Methodology\Revision\High_order_test, I delete all the fields created earlier e1,e2, …, e20. Note if these fields or items are needed, they are still in the file with the same name that was saved under D:\2.Papers\2.CDI-ClimateChange\Paper-1-Methodology\Revision.

5.1 Add fields

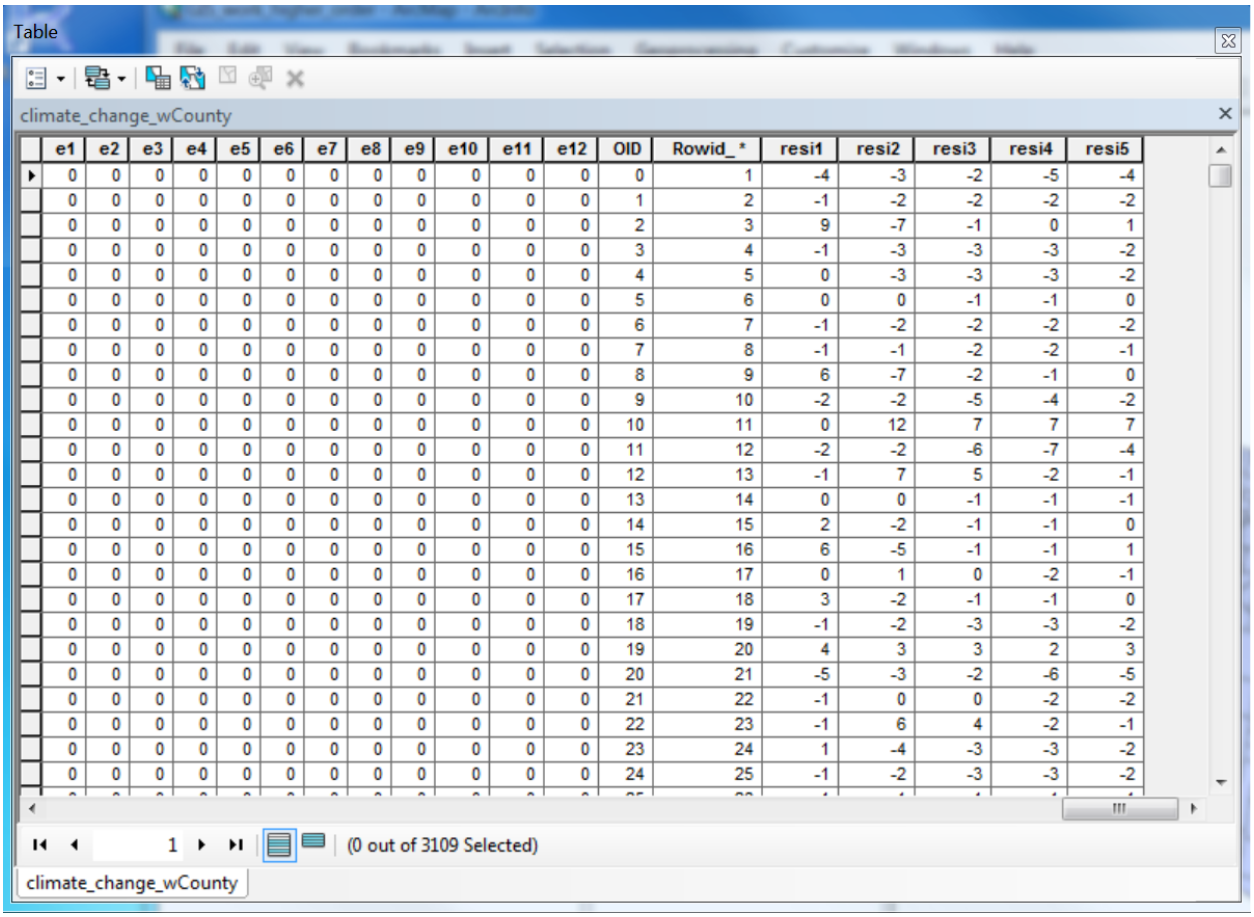
Right click the file → Open Attribute Table → Click the drop-down menu on the top-left corner → click Add Field → Add e1 after Name, choose Float for Type. Similarly add e2, e3, …, e12 to the table.

5.2 Link a dbf file to the shapefile

Right click the file → Joins and Relates → Choose Rowod_ for field, eData1.dbf for table, and automatically Rowid_ will be chosen as the field to base the join on. You will see something like below. However, the OK button is not



available. Do not worry, click the drop-down menu for the field to base the join on and randomly choose another variable. Then choose Rowid_ again, and you will be able to click OK and make the link. Open the table, it will look like this (only the last few columns or fields can be seen)—see the graph below:

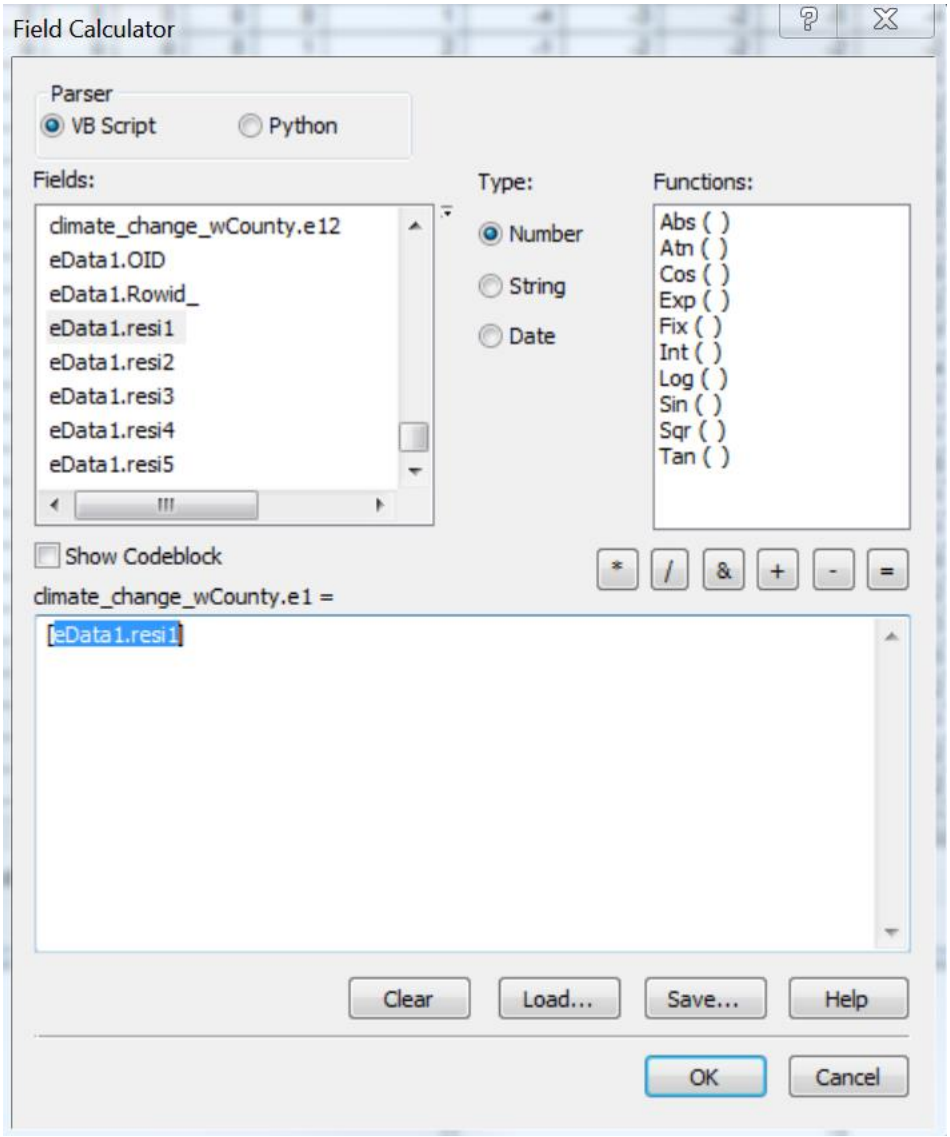



| e1 | e2 | e3 | e4 | e5 | e6 | e7 | e8 | e9 | e10 | e11 | e12 | OID | Rowid_* | resi1 | resi2 | resi3 | resi4 | resi5 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | -4 | -3 | -2 | -5 | -4 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | -1 | -2 | -2 | -2 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 3 | 9 | -7 | -1 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 3 | 4 | -1 | -3 | -3 | -3 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 4 | 5 | 0 | -3 | -3 | -3 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 6 | 0 | 0 | -1 | -1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 7 | -1 | -2 | -2 | -2 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 7 | 8 | -1 | -1 | -2 | -2 | -1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 8 | 9 | 6 | -7 | -2 | -1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 9 | 10 | -2 | -2 | -5 | -4 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 10 | 11 | 0 | 12 | 7 | 7 | 7 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 12 | -2 | -2 | -6 | -7 | -4 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 13 | -1 | 7 | 5 | -2 | -1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 14 | 0 | 0 | -1 | -1 | -1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 15 | 2 | -2 | -1 | -1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 15 | 16 | 6 | -5 | -1 | -1 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 16 | 17 | 0 | 1 | 0 | -2 | -1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 17 | 18 | 3 | -2 | -1 | -1 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 18 | 19 | -1 | -2 | -3 | -3 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 19 | 20 | 4 | 3 | 3 | 2 | 3 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 20 | 21 | -5 | -3 | -2 | -6 | -5 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 22 | -1 | 0 | 0 | -2 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 23 | -1 | 6 | 4 | -2 | -1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 23 | 24 | 1 | -4 | -3 | -3 | -2 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 24 | 25 | -1 | -2 | -3 | -3 | -2 |

5.3. Assign values to e1, e2, …, e12.

Right click Field e1 → choose Field Calculator → put eData1.resi1 into the box (see below) → click OK.

Before assigning values to e2, better to go to the shapefile → Right click → Joins and Relates → Remove Join(s) → Remove All Joins(s). Then repeat 5.2 to link a dbf file (this time eData1.dbf) to the shape file. Note in the box choose eData2.resi1. Then right click Field e2 and complete the Field Calculator step. To make residual comparison consistent and comparable, for all e1, e2, …, e12, we choose resi1 (residuals at time 1) among the five fields that are available.

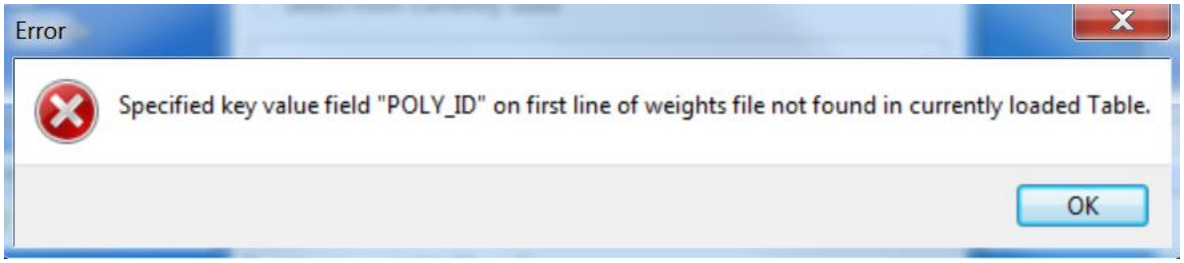



6 sasa

**Step 6. Calculate Moran's I of the residuals for all the 12 connectivity definitions.**

6.1 Add "POLY_ID" to the shapefile

In ArcGIS, open the Attribute Table, and add a field named POLY_ID in the format of "Long Integer". Then right click the added field, choose Field Calculator, choose Rowid_ in the box for POLY_ID. If we do not perform this, you will see the error message below in GeoDa:

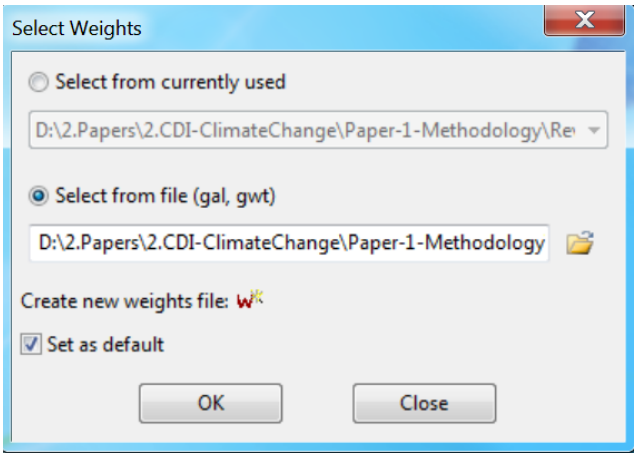



## 6.2 Calculate a bunch of GAL files for Moran's I calculation

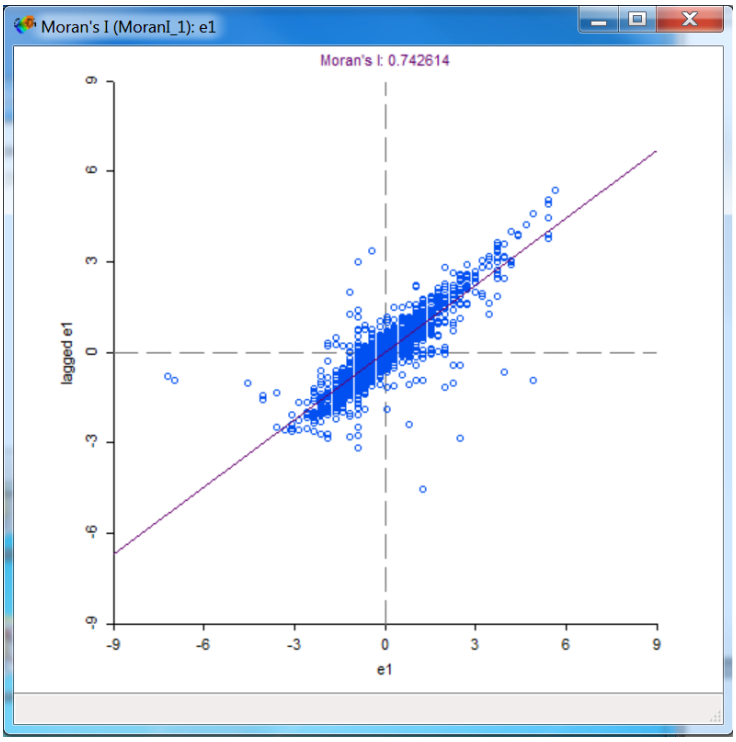
Reopen GeoDa. Note You cannot use ArcGIS' ArcToolbox to calculate the Moran's I here as there are no options for 2nd or higher order connectivity neighborhoods. Follow the same process as in Step 2, generate 12 GAL files that are to be used to calculate Moran's I values, and name them as MoranI_1.GAL, MoranI_2.GAL, …, MoranI_12.GAL. Also check the button for Include lower orders.

## 6.3 Calculate Moran's I and Z scores

Go to GeoDa and click the button  , you will see a window like this below:




Choose the correct spatial weights file we created in 6.2—e.g., MoranI_1.GAL for the 1st order connectivity, click OK. Then click Space and choose Univariate Moran's I from the drop-down menu, or click  , and choose Univariate Moran's I. When the window for Variable Settings comes, choose e1—note: depending what weights file (with GAL extension) you have chosen, the residual files (e1, e2, …, e12) should be chosen in line with that choice. Then the Moran's I is given in the diagram as below:




Then right click the Moran's I window → Randomization → 199 Permutations → from the graph you see the Z-value 73.95. Repeat and calculate the Moran's I and Z scores for all the rest residual files.

Table 2. Moran's I and Z scores under different connectivity definitions

| Order # | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Moran's I | 0.7426 | 0.6595 | 0.5545 | 0.4601 | 0.3676 | 0.2593 | 0.1765 |
| Z score | 72.22* | 110.37 | 134.24 | 129.54 | 137.01 | 116.35 | 94.30 |

| Order # | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|
| Moran's I | 0.1139 | 0.0826 | 0.0653 | 0.0519 | 0.0260 | 0.0330 | 0.0239 |
| Z score | 65.79 | 52.34 | 49.33 | 44.20 | 24.42 | 32.75 | 24.34 |

| Order # | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|
| Moran's I | 0.0145 | 0.0022 | -0.0067 | -0.0131 | -0.0178 | -0.0215 |
| Z score | 16.10 | 2.10 | -7.88 | -19.58 | -25.76 | -42.68 |

Note: Under 199 permutations. As this is a random process, the z scores may change slightly each time you calculate it using the permutation procedure.

**Final decision: Choose a connectivity matrix at the order of 16. Though Table 1 indicates that at this order, only the top 6 eigenvectors have their Moran's I bigger than 0.25, we choose the top 10 eigenvectors as predictor variables in the mixed model. For this model, the residual Moran's I is 0.0015 (z is around 2.10, and the lowest is 1.95 and highest 2.47).**

**Below is the SAS code to perform LTM analysis:**

```
/*
Date: November 6-16, 2014
Code purpose: perform eigenvector analysis to screen out spatial
autocorrelation
Author: Li An
Aknowledgement: Griffith and Chun for the help on eigenvector method
*/

* The input data file ccmatrix.csv is created by adding one row of C1--C3109
to the file created in R using "write.table(cc.mat, file="ccmatrix.csv", sep
= ",", row.names=FALSE, col.names=FALSE)";

* Step 1: import the binary (0/1) matrix generated in R;
proc import datafile="D:\2.Papers\2.CDI-ClimateChange\Paper-1-
Methodology\Revision\High_order_test\ccmatrix16.csv"
      out= conn replace;
run; * Change the directory according to where the file ccmatrix16.csv is
saved;

* Step 2: Construct the MCM and generate the eigen values and eigen vectors;
proc iml;
      use conn;
      read all var _ALL_ into C;
      ones=j(3109,1,1);              * Define a column vector of 1's with 3109
rows;
      I3109=I(3109);                 * Define a 3109 x 3109 identity matrix;
      B=ones*ones`/3109;             * Define an intermediate matrix for
calculating the M matrix;
      M=I3109-B;
      MCM=M*C*M;                     * This step takes 1:19.53;
      *eigenvec=eigvec(MCM);  * Alternative way for calculating eigenvectors.
This step takes 3:06.24;
      *eigv=eigval(MCM);             * Alternative way for calculating
eigenvalues;
      call eigen( val, evec, MCM ) vecl="lvec"; * Calculate both eigenvectors
and eigenvalues in one command;
      * Note all the 3109 eigenvectors are laid out from left to right
according to their corresponding
```

```
        eigenvalues: the eigenvector with the highest eigenvalue is the first
column, then the one with
       the second highest eigenvalue, and so on. But within each eigenvector,
the numbers are listed
        from top to down according to the Polygon ID (here Rowid_ as we used
it to generate the GAL and
            subsequent spatial weights metrics;

    create Eig_value from val[colname='eigen']; * Here create an eigen value
dataset that can be used next step;
    append from val;
      close Eig_value;

    create Eig_vec from evec; * Here create an eigen vector dataset that can
be used next step;
    append from evec;
      close Eig_vec;

      Sum_C = sum(C);           * This returns a certain number, which is
used in Step 3 below;
      print Sum_C;
quit;

/*
* Step 3: Calculate the Moran's I and adjusted eigenvalues according to Dan's
suggestion;
data Moran_I;
      set Eig_value;
      MI = eigen*3109/3546650; * See the docmentation file;
      adj_eigen=eigen/930.3354;
run;
*Outcome: Depending on the input csv file, a varying number of records with
Moran's I > 0.25 or with adjusted
eigenvalue > 0.25;
*/

*Step 4: Choose a set of eigenvectors as predictor variables;

* 4.1 Only keep a subset of eigenvectors;
data eigen_regressors;
      set Eig_vec;
      keep col1-col100; * Depending on how many we want to keep, we can vary
the number;
run;

* 4.2 Add Rowid_ for later data merge;
data eigen_regressors2; *Add a variable called Rowid_ as the merge ID later;
      set eigen_regressors;
      Rowid_ = _n_;     * Note that _n_ is an internal counter. We can do
this as the numbers under
      each eigenvector are listed in an ascending order by Rowid_ as
mentioned earlier;
run;

*From here I need to join the dataset eigen_regressors to the LTM dataset;
```

```
*Step 5: Import the web search and ground data (the same as the code before
using eigenvector method);

/*
proc import datafile="D:\2.Papers\2.CDI-ClimateChange\CC-GW-
Models\MLM\CC_Classified_5_dates_FLOW.xls"
      out= Classified_Data replace;
run; *The path within the quotation marks is where the datafile is saved. The
same for below;

proc import datafile="D:\2.Papers\2.CDI-ClimateChange\CC-GW-
Models\MLM\socioDemoData_wStateData.dbf"
      out= Socio_Demo_Data_State replace;
run;

proc import datafile="D:\2.Papers\2.CDI-ClimateChange\CC-GW-
Models\MLM\Socio_demo_data.xlsx"
      out= Socio_Demo_Data_Cty replace;
run;
*STATEFP COUNTYFP COUNTYNS CNTYIDFP are defined differently in the two input
Excel files--The data step below could
be removed if the data are processed well. But not harmful if we keep;
Data Socio_Demo_Data_State;
      set Socio_Demo_Data_State;
      keep CNTYIDFP sex255212 RHI125212 inc910212 inc110212 rtn131207;
run;

*The sort procedure is necessary for merging the three datasets;
proc sort data=Socio_Demo_Data_State; by CNTYIDFP; run;
proc sort data=Socio_Demo_Data_Cty; by CNTYIDFP; run;
proc sort data=Classified_Data; by CNTYIDFP; run;

data CompData; *The three imported datasets are mergedm using CNTYIDFP as the
key or unique ID;
   merge Classified_Data Socio_Demo_Data_Cty Socio_Demo_Data_State;
   by CNTYIDFP;
run;

%let numOfMSVariables=5; *This is a global variable that can be cited later.
We have data at 5 times;

Data wCounty;
      set CompData;
            whitepct=white;*white is an option in the SAS reg prodecure, so
better to use another name;

      *First education data;
      array msEduDist{&numOfMSVariables} FLO111111C3 FLO030412C3 FLO070112C3
FLO110412C3 FLO030513C3; *The set of rank variables from edu websites;
      array msEdu (*) msEdu1-msEdu&numOfMSVariables; *The variables we want
to create and assign values;

      *Then government data;
      array msGovDist{&numOfMSVariables} FLO111111C6 FLO030412C6 FLO070112C6
FLO110412C6 FLO030513C6; *The set of rank variables from gov websites;
      array msGov (*) msGov1-msGov&numOfMSVariables; *The variables we want
to create and assign values;
```

```
      do i = 1 to &numOfMSVariables;
            msEdu[i]=msEduDist{i};
            msGov[i]=msGovDist{i};
            drop i;
      end;

      Pop_density=N10POP/(BG_Area/1000000);
      if D_Percent=0 then D_Percent=.; *Why: D_Percent=0 means no data (an
artifact of data export in ArcGIS). If
                                                      D_Percent is used in
reg analysis later, all no data records will be
                                                      eliminated because
dot (.) here means no data;
      if R_Percent=0 then R_Percent=.;
      if O_Percent=0 then O_Percent=.;

      RepubPercEvan=Republican1/Total;
      DemocraticPercEvan=Democratic/Total;
      *drop MS06102011--MS09032012;
      drop statefp countyfp namelsad oid1 id geoid_1 countyfp_1 name_1
countyna_1 state_1 geoid fid_1 statefp_1
            countyns_1 cntyidfp_1 cntyidfp_2 cntyidfp_3 countyname;
Run;

*Sample a subset of data--This is not necessary if we use the eigenvector
approach by Griffith. But keep it and change the % to 100%;

Data sampleCntydata; *This step takes a random xx% to avoid spatial
autocorrelation.
      And this dataset is primarily used in the OLS reg ananlysis;
      set wCounty;
      where (selector <= 1.00); *It used to be 0.12. With incomplete
political data, it is increased to 25% to get
      a bigger sample;
      array MSEduRank {&numOfMSVariables} msEdu1-msEdu&numOfMSVariables;
      array MSGovRank {&numOfMSVariables} msGov1-msGov&numOfMSVariables;

      AvgEduRank=sum (of MSEduRank{*})/&numOfMSVariables;
      AvgGovRank=sum (of MSGovRank{*})/&numOfMSVariables;

      drop BG_Count--BG_StdDev date FIPS_State FIPS_County FIPS County_Name
Date1  POSTALCODE;
run;
*/
* Here we combine the eigen vectors and the above sampleCntydata;

*Step 6: Merge the two files: one with selected eigenvectors, one from cyber
and ground data;

* 6.1 Sort the sampleCntydata by Rowid_ in an ascending order first;
proc sort data=sampleCntydata;
      by Rowid_;
run;

* 6.2 Merge the two datasets by Rowid_;
data cntyEigenData;
```

```
      merge sampleCntydata Eigen_regressors2;
      by Rowid_; * At Step 4 we added Rowid_ to Eigen_regressors2 for use
here;
run;

*Step 7: Find out what eigenvectors should be used as predictors through
stepwise regression.
Accoring to Griffith, do stepwise regression to select appropriate
eigenvectors;
/*
proc reg data=cntyEigenData;
      model msEdu1 = col1--col100/selection=stepwise;
run;
* Later I decided to use all top xxx ones with the biggest eigenvalues;
*/

* Step 8: Produce a long-form dataset--data at different times are stacked
vertically;

%let dataTimeSpan=5;
%let BeginTime=1;

data sampleCntydataLTM; *Make the data format change from short form to long
form;
      set cntyEigenData;
      array MSEdu{*} MSEdu&BeginTime-MSEdu&dataTimeSpan;
      array MSGov{*} MSGov&BeginTime-MSGov&dataTimeSpan;
      counter=&dataTimeSpan;

      do Stage=1 to counter by 1;
            Edu=MSEdu{Stage};
            Gov=MSGov{Stage};
            Stagesq=Stage*Stage;
            output;
      end;  * As a result, the number of records in the dataset is 15545 =
3109 x 5;
run;

* Step 9: Do regression with the appropriate eigenvectors chosen in Step 5
plus other variables;

/*
* M1 is not presented in the paper to save space;
* M1: Intercept plus time--with and without eigen vectors;
proc mixed data = sampleCntydataLTM noclprint covtest method=ml;
      class CNTYIDFP;
      model edu = Stage col1--col100/solution ddfm=bw notest;* Take out col1
col9 col12 to test their effect;
      random intercept Stage/sub=CNTYIDFP type=cs;
      repeated/sub=CNTYIDFP type =cs r rcorr;
      title "M1. Intercept and slope: random coefficient model";
run;
*/

* 8.1--M1: Intercept, time, and time square--with eigen vectors--this is new
M1 (former M2);
proc mixed data = sampleCntydataLTM noclprint covtest method=ml;
```

```
      class CNTYIDFP;
      model edu = Stage Stagesq col1--col100/solution ddfm=bw notest;
      random intercept Stage Stagesq/sub=CNTYIDFP type=cs;
      repeated/sub=CNTYIDFP type =cs r rcorr;
      title "M1: Intercept, slope, slope-square, and eigenvectors: random
coefficient model";
run;

/*
* M3:  Intercept, time, and covariates--with eigen vectors;
proc mixed data = sampleCntydataLTM noclprint covtest method=ml;
      class CNTYIDFP;
      model edu = Stage RepubPercEvan Pop_density POP_urban SummerDays
MaxDrySpel N10_16over medhhldinc
      Stage*RepubPercEvan Stage*Pop_density Stage*POP_urban Stage*SummerDays
Stage*MaxDrySpel Stage*N10_16over
      Stage*medhhldinc col1--col100
      /solution ddfm=bw notest;
      random intercept Stage/sub=CNTYIDFP type=cs g;
      repeated/sub=CNTYIDFP type = cs  r rcorr;
      title "M3. Intercept, slope, and covariates: random coefficient model";
run;
*/

* 8.2--M2:  Intercept, time, time square, and covariates--with eigen vectors-
-this is new M2 (former M4);

*Note: comment out the three RepubPercEvan terms when calculating the
residuals and outputing them to the file
named test. When running the code for parameter values, restore the three
RepubPercEvan related terms;
proc mixed data = sampleCntydataLTM noclprint covtest method=ml;
      class CNTYIDFP;
      model edu = Stage Stagesq RepubPercEvan Pop_density POP_urban
SummerDays MaxDrySpel N10_16over medhhldinc
       Stage*RepubPercEvan Stage*Pop_density Stage*POP_urban Stage*SummerDays
Stage*MaxDrySpel Stage*N10_16over
            Stage*medhhldinc col1--col10
      Stagesq*RepubPercEvan Stagesq*Pop_density Stagesq*POP_urban
Stagesq*SummerDays Stagesq*MaxDrySpel
            Stagesq*N10_16over Stagesq*medhhldinc /solution outp=test ddfm=bw
notest; * I added outp=test to report the predicted values and residuals Nov.
8, 2014;
      random intercept Stage Stagesq/sub=CNTYIDFP type=cs g;
      repeated/sub=CNTYIDFP type = cs  r rcorr;
     title "M2: Intercept, slope, slope-square, eigenvectors, and covariates:
random coefficient model";
run;

/*
* Step 9: Check whether M4's rediduals are normally distributed;
proc stdize data= test out=fig2;
  var resid;
run;
proc univariate data = fig2 noprint; * Here we examine whether the
standardized residuals are normally distributed;
  var resid;
```

```
  qqplot /href = 0 vref = 0 ;
  title "QQ plot for M4";
run;
*/

* Step 10: Output the residuals so that their Moran's I can be calculated in
GeoDa;

* 10.1. The above test dataset is a dataset with 3109x5 = 15545 rows. So I
need to convert it back to a dataset with
only 3109 records;
data Resi_data;
  set Test;
      do i=1 to 15545 by 5;*choose 1,6,11, ..., 15541 b/c the data were
organized this way:
      county's data at times 1, 2, 3, 4, and 5;

      *The arrays are reusable containers to temporarily hold the data at
five times;
            array resids {5} (0,0,0,0,0);

            if _n_=i then do;
                  resids{1}=resid;
                  end;
            else if _n_=i+1 then do;
                  resids{2}=resid;
                  end;
            else if _n_=i+2 then do;*The third time;
                   resids{3}=resid;
                  end;
            else if _n_=i+3 then do;*The fourth time;
                  resids{4}=resid;
                  end;
            else if _n_=i+4 then do; *The last time period (here 2000 or the
end of 1990s);
                  resids{5}=resid;
                  end;

            if _n_=i+4 then   do;
                  flag=1;

      resi1=resids{1};resi2=resids{2};resi3=resids{3};resi4=resids{4};resi5=r
esids{5};
            end;
      end;

      if flag=1 then output; *only output records 5, 10, 15...b/c other
records flag<>1;
      keep Rowid_ resi1--resi5;
run;
/**/
* 10.2. A test to find out the 10% and 90% quantitles of the residuals in
datafile Resi_data;
proc univariate  data =Resi_data;
      var resi1;
run; * The 10% and 90% quantitles are -5.6559 and 6.4042;
```

```
data testRan;
      do i = 1 to 50;
      x = -5.6559 +(6.4042 + 5.6559)*ranuni(123);
      output;
      end;
run;


* 10.3. Replace no data dots (.) with random numbers that follow the
distribution of residuals;
data Resi_data;
      set Resi_data;
      if resi1=. then do;
            resi1=-5.6559 +(6.4042 + 5.6559)*ranuni(123);
      end;
run;

* 10.4. Below we output then above dataset Resi_data to a dbf file that can
be used in ArcGIS;
filename eData 'D:/2.Papers/2.CDI-ClimateChange/Paper-1-
Methodology/Revision/High_order_test/eData.dbf';
proc dbf db5=eData data=Resi_data;
run;
```