Government 1016: Spatial Models

LAB EXERCISE 8: “Visualizing with STARS”

Objectives:

- Descriptive space time clustering indices
- Interpreting Markov transitions matrix

Estimated time: 4 hours

The authors of STARS – Prof. Sergio Rey and others at San Diego State University, San Diego, CA and Arizona State University, Tempe, AZ describe Space-Time Analysis of Regional Systems (STARS) as follows:

“Space-Time Analysis of Regional Systems (STARS) is an open source package designed for the analysis of areal data measured over time. STARS brings together a number of recently developed methods of space-time analysis into a user-friendly graphical environment offering an array of dynamically linked graphical views. It is intended to be used as an exploratory data analysis tool.”

They also note that:

“STARS has been implemented using the Python scripting language. It consists of a series of Python modules which can be used as stand alone components or together in an interactive user interface.”

STARS software can be downloaded from **http://regionalanalysislab.org/?n=Download** and there is some (not very much) documentation at **http://regionalanalysislab.org/?n=StarsPubs** and **http://regionalanalysislab.org/?n=Screenshots**.

The Mac version works on the lab computers. Note that STARS will work only on Windows XP so it will not work on PCs in the lab (which have Windows 7). STARS will only work on the lab Macs. If must work on a PC then use PySAL. can use PySal to conduct this analysis if are familiar with using Python. For this go to the download page:
http://code.google.com/p/pysal/
Note that to run pysal need to make sure have Python 2.5 or later, Numpy 1.5 or later and Scipy 0.7 or later installed on r computer. Note that there is a PC and Mac version available.
Documentation is here: **http://pysal.geodacenter.org/1.3/index.html**
Installation instructions are here: **http://pysal.geodacenter.org/1.3/users/installation.html**

1.0 Space-Time Analysis

There are two components to STARS: A project maker and the software itself. The next section is optional since the projects for county and state based income are already available in the STARSdata folder. may find the next section useful when need to create a project with your own data. All project files have a .prj extension. In the Gov1016\STARSdata folder you will find two STARS projects: county.prj, and state.prj. These projects have time series data on personal income and personal income per capita by county and state. The county data have annual data from 1969-

2006 and the state data have annual figures from 1929-2007. The state data also include data on population for those years. The data were downloaded from the BEA (Bureau of Economic Analysis). The shapefiles for county and state data were obtained from the US Census Bureau.

1.1 STARS Projects (Optional only useful if are creating data – there is already a project available in the STARSdata folder)

1. To launch Project Maker double click on the "Project Maker" application.

This leads to the main screen for Project Maker which contains four main menus: File, Data, Tables, Plot.

2. Select main menu File, then "Create New STARS Project." You will be prompted to choose which type of file that you wish to use as the base data for the new project. r two choices are ArcView and CSV. If r project has an associated shapefile, you must choose the ArcView option. can always add data from comma delimited (csv) files later in the project making process. The steps that follow are identical for the CSV option except you cannot plot the map.
3. For this example choose: ArcView. You will then be prompted to choose a file that contains r base data. Base data for ArcView projects are found in *.dbf files. Navigate to r data folder and choose the dbf file associated with the shapefile that you have time series data for. In this case I used county48 for county level data and state48 for state level data.

4. You will next be prompted to name r new project. If you leave this entry blank, the prefix of the base data file will be the name of r project (i.e. county48 or state48).

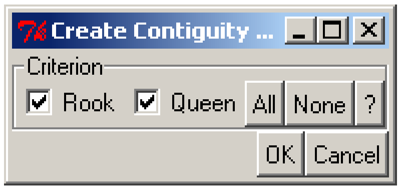
For this example I named the projects, county (and state). Click "OK". Next you will have to declare the time series information for r project. First you will be prompted for the type of series.

5. Both the examples I used had annual data, so check the "Annual" button and click "OK". Next, you must provide the start year and the end year for r time series. So place 1969 and 2006 in their corresponding entry spaces for county and 1929 and 2007 for state, and click "OK".

There are several canned types of time-series projects in STARS. Here is a brief description:

- Decadal: Provide the start year and the end year when creating the project.
- Annual: Provide the start year and the end year when creating the project.
- Quarterly: Provide the start quarter, start year, end quarter, and end year.
- Monthly: Provide the start month, start year, end month, and end year.
- Irregular: For all other cases, it is best to use the irregular time-series.

6. Just provide the number of time intervals in r project.
7. will now encounter a dialogue box where must choose a field from r original data to use as the names for r cross-sectional units. Depending on the shapefile there will be empty fields in this view. In the case of state, you only need to select "FIPS" for the "Unique Field" and then NAME for the operational name field and then click "OK".

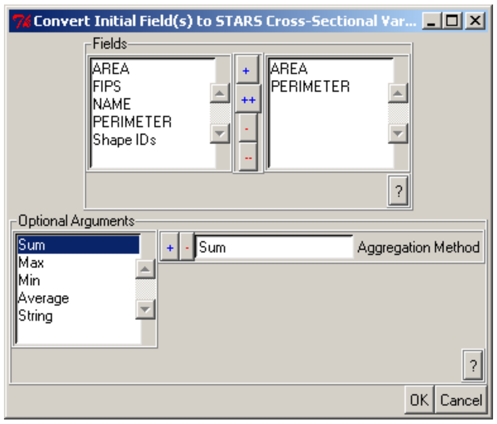

Create Contiguity ...
Criterion
☑ Rook ☑ Queen

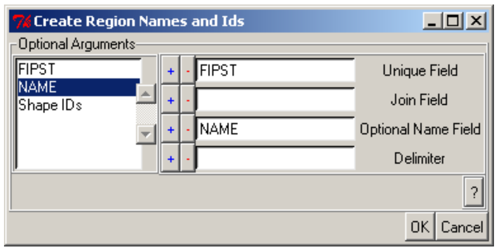
This will associate any shape in r shapefile with its corresponding name. The next step involves selecting what type (if any) weights matrices you would like for spatial proximity. Select the type you want (rook or queen or all) and click "OK." The continuity matrices will next be created. Once this is done you can start placing data in the project.

There are several methods that can be used to place data in STARS projects. They fall under the following categories: Convert, Merge, Join. There are also three different types of data:

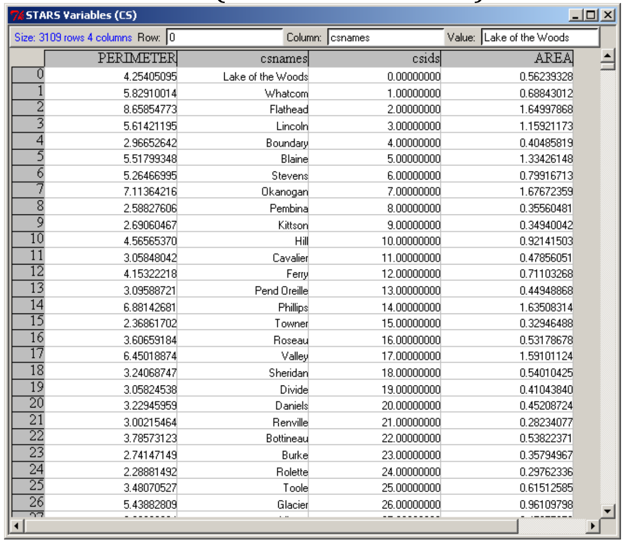
- CS Cross-Sectional Data
- TS Time-Series Data
- CSTS Cross-Sectional/ Time-Series (Panel) Data.

8. Select main menu Data, then Variable, then Convert Base Data to CS. will encounter a dialogue box with all of the base data Fields in one listbox, and a series of Optional Arguments in the other as shown in the screenshot. I chose "AREA" and "PERIMETER" as variables that I wanted to add to the STARS project called county. To do so, highlight "AREA" in the listbox and click the "+" button. Highlight "Sum" as r Optional Argument, and click "OK". (See screen on the side). Youhave just created a new "CS" variable for r STARS project.



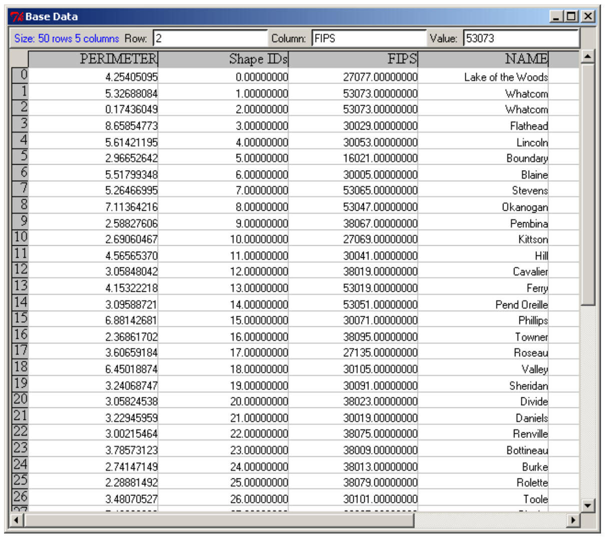


9. To view your results you can use the appropriate table by clicking on main menu Tables and then CS. (See screens below).



Size: 3109 rows 4 columns Row: 0 Column: csnames Value: Lake of the Woods

	PERIMETER	csnames	csids	AREA
0	4.25405095	Lake of the Woods	0.00000000	0.56239328
1	5.82910014	Whatcom	1.00000000	0.68843012
2	8.65854773	Flathead	2.00000000	1.64997868
3	5.61421195	Lincoln	3.00000000	1.15921173
4	2.96652642	Boundary	4.00000000	0.40485819
5	5.51799348	Blaine	5.00000000	1.33426148
6	5.26466995	Stevens	6.00000000	0.79916713
7	7.11364216	Okanogan	7.00000000	1.67672359
8	2.58827606	Pembina	8.00000000	0.35560481
9	2.69060467	Kittson	9.00000000	0.34940042
10	4.56565370	Hill	10.00000000	0.92141503
11	3.05848042	Cavalier	11.00000000	0.47856051
12	4.15322218	Ferry	12.00000000	0.71103268
13	3.09588721	Pend Oreille	13.00000000	0.44948868
14	6.88142681	Phillips	14.00000000	1.63508314
15	2.36861702	Towner	15.00000000	0.32946488
16	3.60659184	Roseau	16.00000000	0.53178678
17	6.45018874	Valley	17.00000000	1.59101124
18	3.24068747	Sheridan	18.00000000	0.54010425
19	3.05824538	Divide	19.00000000	0.41043840
20	3.22945959	Daniels	20.00000000	0.45208724
21	3.00215464	Renville	21.00000000	0.28234077
22	3.78573123	Bottineau	22.00000000	0.53822371
23	2.74147149	Burke	23.00000000	0.35794967
24	2.28881492	Rolette	24.00000000	0.29762336
25	3.48070527	Toole	25.00000000	0.61512585
26	5.43882809	Glacier	26.00000000	0.96109798



Size: 50 rows 5 columns Row: 2 Column: FIPS Value: 53073

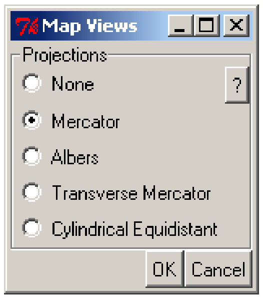
	PERIMETER	Shape IDs	FIPS	NAME
0	4.25405095	0.00000000	27077.00000000	Lake of the Woods
1	5.32688084	1.00000000	53073.00000000	Whatcom
2	0.17436049	2.00000000	53073.00000000	Whatcom
3	8.65854773	3.00000000	30029.00000000	Flathead
4	5.61421195	4.00000000	30053.00000000	Lincoln
5	2.96652642	5.00000000	16021.00000000	Boundary
6	5.51799348	6.00000000	30005.00000000	Blaine
7	5.26466995	7.00000000	53065.00000000	Stevens
8	7.11364216	8.00000000	53047.00000000	Okanogan
9	2.58827606	9.00000000	38067.00000000	Pembina
10	2.69060467	10.00000000	27069.00000000	Kittson
11	4.56565370	11.00000000	30041.00000000	Hill
12	3.05848042	12.00000000	38019.00000000	Cavalier
13	4.15322218	13.00000000	53019.00000000	Ferry
14	3.09588721	14.00000000	53051.00000000	Pend Oreille
15	6.88142681	15.00000000	30071.00000000	Phillips
16	2.36861702	16.00000000	38095.00000000	Towner
17	3.60659184	17.00000000	27135.00000000	Roseau
18	6.45018874	18.00000000	30105.00000000	Valley
19	3.24068747	19.00000000	30091.00000000	Sheridan
20	3.05824538	20.00000000	38023.00000000	Divide
21	3.22945959	21.00000000	30019.00000000	Daniels
22	3.00215464	22.00000000	38075.00000000	Renville
23	3.78573123	23.00000000	38009.00000000	Bottineau
24	2.74147149	24.00000000	38013.00000000	Burke
25	2.28881492	25.00000000	38079.00000000	Rolette
26	3.48070527	26.00000000	30101.00000000	Toole

10. You can then start merging or joining some of the available time series data. Click main menu Data, then Variable, then Merge and then finally CSTS Data.
11. You will encounter a file browser that should identify only files with a *.csv extension. Navigate to the directory where you have the data to be merged – they must be saved in csv

format. Also, they should be ordered in the same way as r county shapefile and only one text header is allowed which is the name of the variable (pinc for example).

12. Merging data is easy on the front end, but requires that the data is ordered in the same way as the project you just created.
13. If you have only time series data you can click main menu Data, Variable, Merge and then TS Data and add the file which would be for the entire country over the same time period.

To join data instead of merging you will encounter a file browser that should identify only files with a *.csv extension. You should navigate to the data directory where you should choose the file and then will be asked to choose the appropriate fields to match. This should be a unique key.

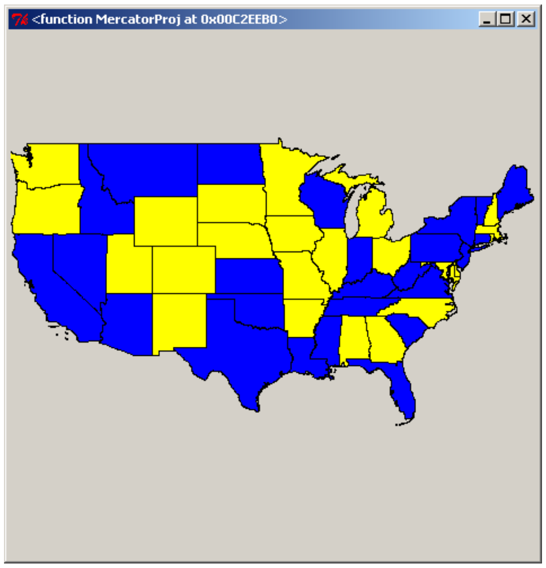

Map Views

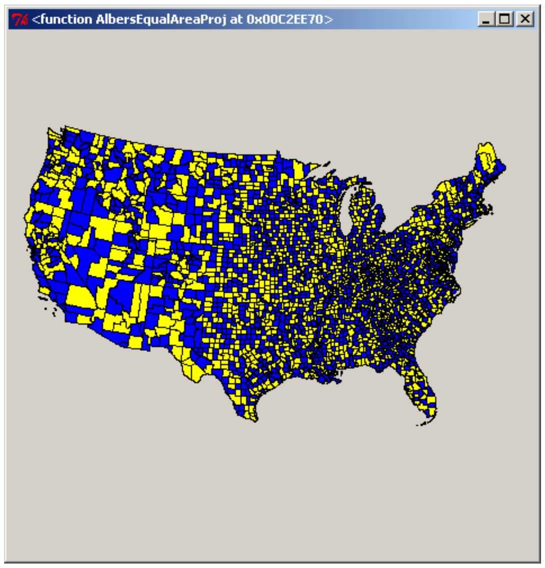
Projections

- None
- Mercator [✓]
- Albers
- Transverse Mercator
- Cylindrical Equidistant

Once you have added all of the data and matrices you want in r new STARS project, you can then save the file. If r project has a shapefile, you must plot r map before you can save the project.

14. Select main menu Plot, then "Plot Map," you will be prompted to choose the type of projection you want for r map. Choose a type (I chose Mercator) and select "OK." Lastly, select main menu File, and then "Save Project."





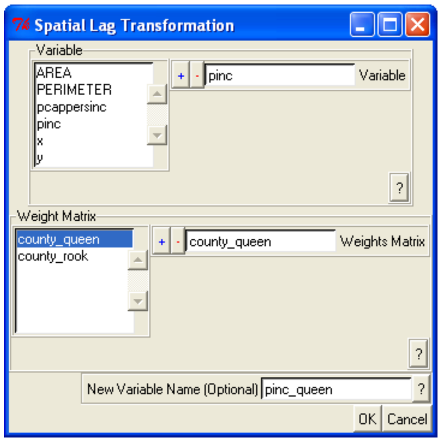
State level data ***County level data***

STARS does not require that r project contain a map. A comma-delimited file *.csv can be used to create a project. However all of the mapping utilities in STARS will be unavailable. The remaining steps are the same, except for the plotting feature.

Some shapefiles contain cross-sections that are made up of a series of islands. For example, several counties can be made up of a major land mass and several islands. I used a newly created "ALLID" field for a unique identifier which was then aggregated to combine all shapes with the same "NAME" to form a single cross-sectional unit. In essence, the "NAME" field was not unique, but instead of joining fields or using a separate unique field to keep all of the initial shapes separate, I aggregated to the number of unique values in the field. When an aggregation occurs in the base data it is necessary to provide optional arguments when converting base data to STARS variables. If a new cross-sectional unit is made up of a series of islands, each separate shape will have attributes that need to be aggregated as well. r choices include:

- Sum: Adds the values for all the shapes that make up each cross-section. We used this in county for the field "AREA". This makes sense because we are aggregating several shapes into one, and we would probably want the total area for each cross-section.
- Max: Returns that maximum value for the variable in question for all shapes that make up each cross-section.
- Min: Returns that minimum value for the variable in question for all shapes that make up each cross-section.

- Average: Returns that average value for the variable in question for all shapes that make up each cross-section.



Spatial Lag Transformation

Variable: AREA, PERIMETER, pcappersinc, pinc, x, y

Variable: pinc

Weight Matrix: county_queen, county_rook

Weights Matrix: county_queen

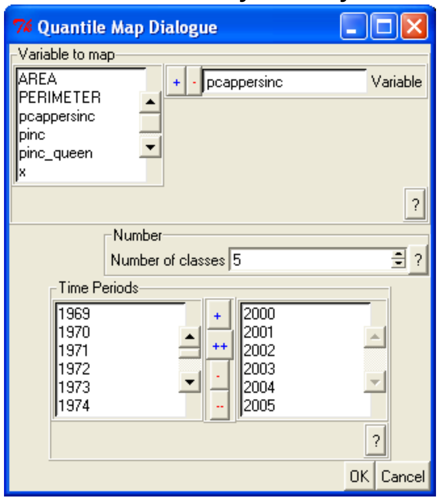
New Variable Name (Optional): pinc_queen

OK Cancel

1.1 Visualization in STARS

1. First, in main menu **File** and then **Open** and select **county.prj** which is in the Gov1016\STARSdata folder in STARS. This will take some time and generate a project summary in the output window.

This project contains data on per capita income in the lower 48 US states on an annual basis from **1969-2006** by county. The project has 6 variables, 4 of which are purely cross-sectional (CS), 2 being cross-sectional time-series (CSTS) and no pure time-series (TS) variable. In addition the project has two matrices which are the contiguity matrices using rook and queen criteria. The two CSTS variables are **personal income** and **personal income per capita** by county from 1969-2006. Perimeter and Area are by county. The other two variables indicate centroids.



Quantile Map Dialogue

Variable to map: AREA, PERIMETER, pcappersinc, pinc, pinc_queen, x

Variable: pcappersinc

Number of classes: 5

Time Periods: 1969, 1970, 1971, 1972, 1973, 1974

Selected: 2000, 2001, 2002, 2003, 2004, 2005

OK Cancel

2. You may create new variables such as spatial lags by selecting main menu **Data**, then **Create** and then **Spatial lags** as shown in the screen.
3. You may also look at the histograms of the contiguity matrices by selecting **Data** and then **Matrix** and then **Characteristics**.
4. <u>Next</u>, try mapping some of the data by selecting main menu **Visualization** and then **Map**, **Quantile** to get a window as shown in the screen. Use the "+" to add a variable.

You could try some of the other options in the menu as well (Standard deviation for example). may choose to map different points of time over the period.

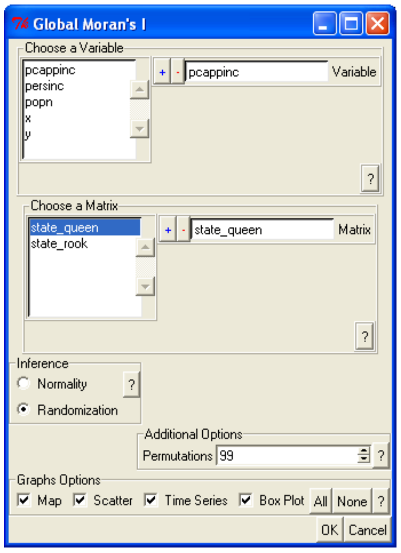
5. If you right click on the map you will get several options including being able to zoom in or out; and interaction.
6. For interaction can try **linking** in which the selection of observations in an origin view (here the map) leads to the highlighting of associated observations in other destination views (such as box plots or histograms or any other type of visualization that have open) using a rectangle created and sized with the mouse. When you release the mouse button, the polygons underneath the selection rectangle are selected and observations associated with these selected polygons are then highlighted in the destination views.

7. The second form of interaction is **brushing**. Here observations are selected in the same way as with linking, however the impact of the selected set is different, and results. For example it would result in a refitting of the global autocorrelation trend in the scatter plot to omit the states selected on the map. This provides insights as to the leverage of the selected states on the level of spatial clustering for that time period.
8. Linking and brushing can also be combined with a third form of interaction referred to as **roaming**. When roaming, the selection rectangle remains on the screen and the user can move it around the origin view. Movement of the selection rectangle creates a new selection set of observations on the origin view to trigger the corresponding interaction signal (brushing or linking) on the destination views.
9. Similar to roaming, linking and brushing can also be combined with traveling. Traveling on an origin view selects observations in a sorted order and triggers linking or brushing on the destination views. This is the equivalent of the map movie you saw in Geoda. Traveling can also be done on a time series view to trigger temporal updating of destination views.

Open the states project and try out some of the other visualization options (there are fewer states than counties so that speeds up the computations considerably).

Question 1 (20 points)

Using the **<u>state</u>** level data create at least: one map, a regional time series, a histogram, a scatterplot and a box plot to test out the visualization capabilities of STARS. You should also use these visuals to tell a "story" that describes income changes over time in the region of r choice. Note that you can do this in GeoDA using the state48_ shapefiles which have the time series data joined to the state shapefile.


Global Moran's I

Choose a Variable: pcappinc, persinc, popn, x, y — Variable: pcappinc

Choose a Matrix: state_queen, state_rook — Matrix: state_queen

Inference: Normality, [✓ Randomization]

Additional Options: Permutations 99

Graphs Options: Map ✓, Scatter ✓, Time Series ✓, Box Plot ✓

2.0 Analysis in STARS

8. In the main menu Analysis, select ESDA, then select first **Moran Global** and then **Moran Local** for one of the variables in state and county as shown. (You could also try Getis or Geary equivalents). Use brushing to see the effects of outliers on the results.

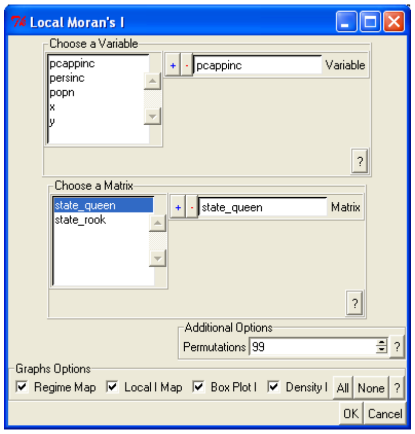
What trends do you notice at the state and county level? may choose to examine at a region, a state or overall trends.

Question 2a (20 points)

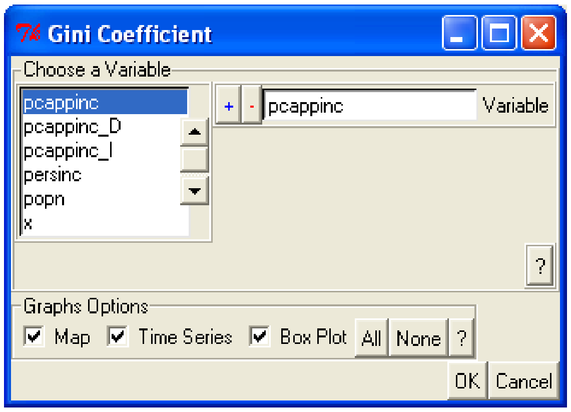
Describe the trends at the state and county level for the variable that you chose in terms of the maps, global and local Moran results and the time path. Do you think the choice of contiguity matrix will make a difference to the results? Compare results (Getis or Moran). (Again this can be done in Geoda as well except the time path).

9. Next, select in the main menu Analysis, Inequality… and then first Classic Gini, followed by spatial Gini and global Theil for the variable of r choice. Compare the time plots that you obtain.

Spatial Gini is considers distribution across a county or states land area. It is 0 if all the income is concentrated in one region and 0 if it is distributed across the country. First, the counties are ranked in descending order according to density of income then the cumulative shares of land are are derived and a corresponding series of cumulative shares of income by county (or state). Then these two measures are plotted against each other to produce a Lorenz curve and the spatial Gini is calculated as the ratio of the area between the uniform distribution line and the Lorenz curve to the area between the uniform distribution line.



Variable: pcappinc
Matrix: state_queen
Permutations: 99
Graphs Options: ☑ Regime Map ☑ Local I Map ☑ Box Plot I ☑ Density I

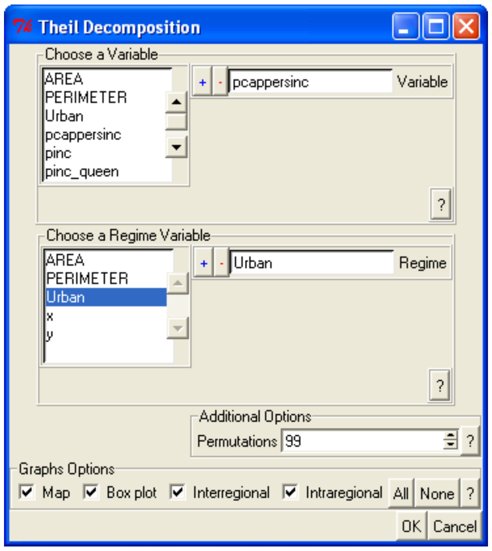


Variable: pcappinc
Graphs Options: ☑ Map ☑ Time Series ☑ Box Plot

Question 2b (20 points)

Describe the time paths for classic Gini, spatial Gini and global Theil index at the county and/ or State level. What do the path trends tell you about how the variable changed over time? (An increase indicates increase in the distribution of the variable) (This is not available in PySAL)

10. might also be interested in the Theil decomposition (go to the main menu Analysis, Inequality… and then Theil Decomp)



Variable: pcappersinc
Regime: Urban
Permutations: 99
Graphs Options: ☑ Map ☑ Box plot ☑ Interregional ☑ Intraregional

Some regime variables include urban county or not (does the county have an urbanized area according to the Census) in the county data and regional location within the US (Northeast, South, and Southwest/ Mountain state) in the state data. If a population can be classified into mutually exclusive and completely exhaustive groups, then Theil's T statistic is made up of two components, the between regime element (T'_g) and the within regime element ($T^w{}_g$). Where the overall $T = T'_g + T^w{}_g$

The between group element of Theil's T can be written as:

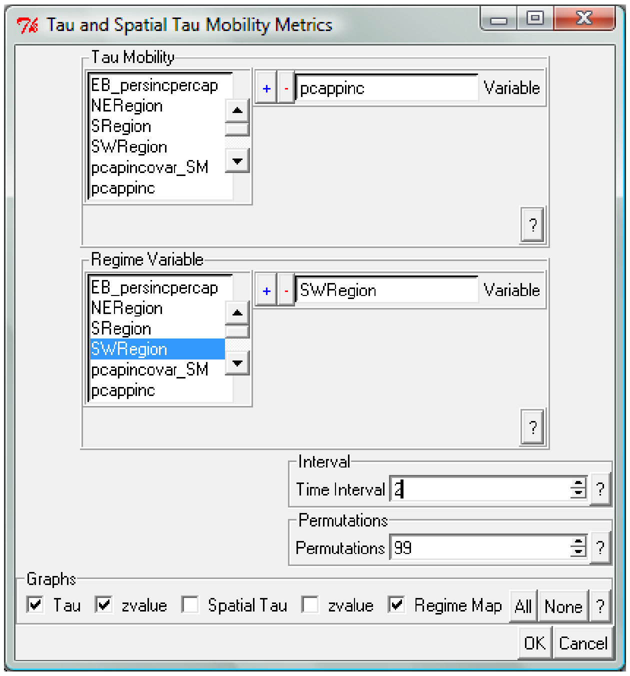
$$T'_g = \sum_{i=1}^{m} \left\{ \left(\frac{p_i}{P} \right) * \left(\frac{y_i}{\mu} \right) * \ln \left(\frac{y_i}{\mu} \right) \right\}$$

where i indexes the regimes, p_i is the population of regime i, P is the total population, y_i is the average income in regime i, and μ is the average income across the entire population.

Look for the trends in interregional versus intraregional Theil indices. What does this indicate about interregional versus intraregional changes in the Theil index of the variable over time?

Question 2c (10 points)

Describe the trends in Theil decomposition at the state and county level by a regime at either the county or state level and attach appropriate graphs. (This is not available in PySAL)



2.1 Tau and Theta: rank based measures

11. Also look at the Kendall's tau (main menu Analysis, Mobility, Tau and Theta).

These are measured by first ranking the locations and then seeing the change in ranks over the two time periods.

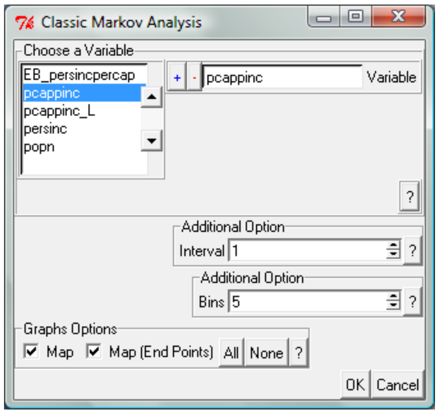
Kendall's $\tau \quad = \quad \dfrac{N_c - N_d}{(n^2 - n)/2}$

where N_c is the number of concordant pairs, and N_d the number of discordant pairs. If all pairs are concordant, then $N_c = (n^2 - n)/2$, $N_d = 0$ and $\tau = 1$, while if all pairs are discordant, then $\tau = -1$. Thus τ shows the extent to which growth is parallel across entire system.

Theta $\theta_{t1-t0} = \dfrac{\Sigma_R \,|\, \Sigma_{i\in R}\, \theta_{i,t1} - \theta_{i,t0} \,|}{\Sigma_i \,|\, \theta_{i,t1} - \theta_{i,t0} \,|}$

$\theta_{i,t1}$: location i's rank at time t

R: a set of exhaustive and mutually exclusive groups of spatial units. θ should be between 0 and 1. With more cohesion, the measure approaches 1. It measures the extent to which growth is competitive across regional partitions.



Question 2d (20 points)

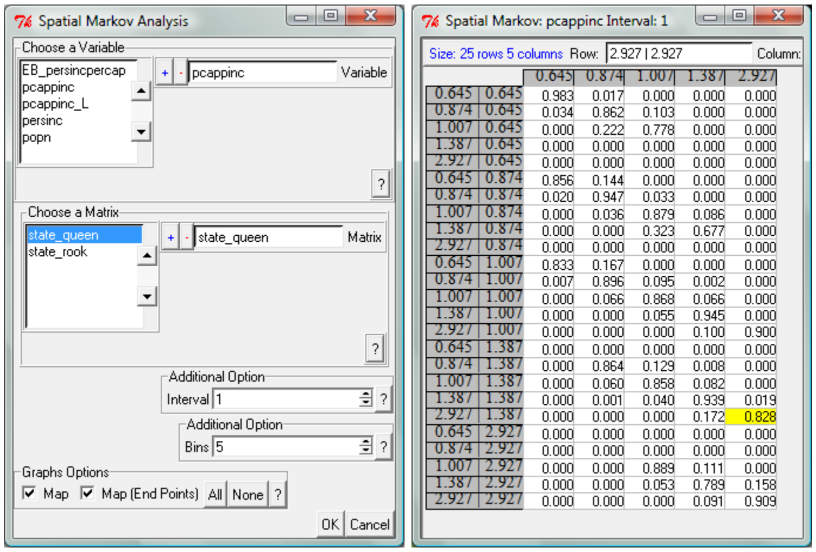
Describe the Tau and Theta trends for one of the variables over time.

2.2 Markov transition matrices

Regular Markov matrices have p(m; h) probability of transitioning from one income class to another over a time interval which assumes transitions are spatially random. Spatial Markov includes conditional transition probabilities.

12. Generate the classic and spatial Markov matrix. (Main menu Analysis, Markov and then classic and spatial).

The cells in transition matrices can be selected to see the corresponding locations on the maps. (See screenshots below).



Choose a Variable
EB_persincpercap
pcappinc
pcappinc_L
persinc
popn

pcappinc Variable

Choose a Matrix
state_queen
state_rook

state_queen Matrix

Additional Option
Interval 1

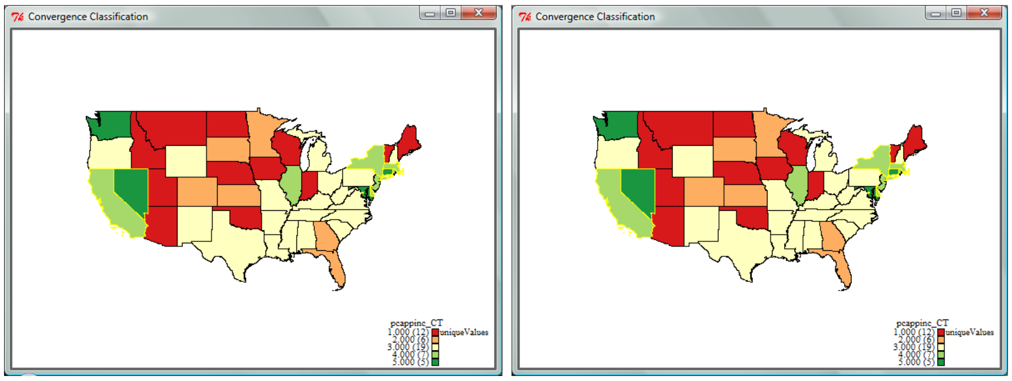
Additional Option
Bins 5

Graphs Options
☑ Map ☑ Map (End Points)

Size: 25 rows 5 columns Row: 2.927 | 2.927 Column:

		0.645	0.874	1.007	1.387	2.927
0.645	0.645	0.983	0.017	0.000	0.000	0.000
0.874	0.645	0.034	0.862	0.103	0.000	0.000
1.007	0.645	0.000	0.222	0.778	0.000	0.000
1.387	0.645	0.000	0.000	0.000	0.000	0.000
2.927	0.645	0.000	0.000	0.000	0.000	0.000
0.645	0.874	0.856	0.144	0.000	0.000	0.000
0.874	0.874	0.020	0.947	0.033	0.000	0.000
1.007	0.874	0.000	0.036	0.879	0.086	0.000
1.387	0.874	0.000	0.000	0.323	0.677	0.000
2.927	0.874	0.000	0.000	0.000	0.000	0.000
0.645	1.007	0.833	0.167	0.000	0.000	0.000
0.874	1.007	0.007	0.896	0.095	0.002	0.000
1.007	1.007	0.000	0.066	0.868	0.066	0.000
1.387	1.007	0.000	0.000	0.055	0.945	0.000
2.927	1.007	0.000	0.000	0.000	0.100	0.900
0.645	1.387	0.000	0.000	0.000	0.000	0.000
0.874	1.387	0.000	0.864	0.129	0.008	0.000
1.007	1.387	0.000	0.060	0.858	0.082	0.000
1.387	1.387	0.000	0.001	0.040	0.939	0.019
2.927	1.387	0.000	0.000	0.000	0.172	0.828
0.645	2.927	0.000	0.000	0.000	0.000	0.000
0.874	2.927	0.000	0.000	0.000	0.000	0.000
1.007	2.927	0.000	0.000	0.889	0.111	0.000
1.387	2.927	0.000	0.000	0.053	0.789	0.158
2.927	2.927	0.000	0.000	0.000	0.091	0.909

Spatial Markov



Convergence maps

Question 2e (20 points)

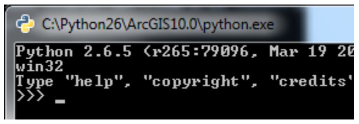
Interpret the two matrices. Contrast the probability of change in income using the Markov transition matrices that you get. Also interpret the LISA Markov matrices.

References and further reading

Readings related to STARS: **http://regionalanalysislab.org/?n=StarsPubs**
Rey, S. J. and L. Anselin, PySAL a Python Library for Spatial Analytical Methods, in Handbook of Applied Spatial Analysis, **http://www.springerlink.com/content/u7r3m73428ukj365**
Readings related to PySal, **http://geodacenter.asu.edu/research/publications/pysal**

Pysal instructions

Note that you can do spatial autocorrelation, smoothing, spatial regressions etc in Pysal which is described in the documentation. Since you can do this in Geoda or R I will not describe it here. Refer to:
http://pysal.geodacenter.org/1.3/users/tutorials/


C:\Python26\ArcGIS10.0\python.exe

```
Python 2.6.5 (r265:79096, Mar 19 20
win32
Type "help", "copyright", "credits'
>>> _
```

Start python from the folder where you installed it. If you have ArcGIS on r machine it will be at C:\Python26\ArcGIS10.0

Load the shapefiles and weights files (*.GAL or *.gwt) from the folder STARSData that plan to use in a folder called examples under the folder pysal in Python26

```
import os
import pysal
help(pysal)

shp = pysal.open('../pysal/examples/state48_percapinc.shp')
poly = shp.next()
type(poly)
len(shp)
shp.get(len(shp)-1).id
polys = list(shp)
len(polys)

help(next)
help(len)
help(poly)
```

Refer to **http://pysal.geodacenter.org/1.3/users/tutorials/dynamics.html** for more detail on the following steps. Note that you cannot do Questions 2b and 2c in this version of Pysal.

```
f = pysal.open("../pysal/examples/state48_percapinc.csv")

f.header
f[0]
f[9]
f[:,8]
import numpy as np

pci = np.array([f.by_col[str(y)] for y in range(1929,2007)])
pci.shape
```

The first row of the array is the per capita income for the first year

```
pci[0, :]
```

In order to apply the classic Markov approach to this series, you first have to discretize the distribution by defining your classes. There are many ways to do this, but here you will use the quintiles for each annual income distribution to define the classes:

```
q5 = np.array([pysal.Quantiles(y).yb for y in pci]).transpose()
q5.shape
```

A number of things need to be noted here. First, you are relying on the classification methods in PySAL for defining our quintiles. The class Quantiles uses quintiles as the default and will create an instance of this class that has multiple attributes, the one you are extracting in the first line is yb - the class id for each observation. The second thing to note is the transpose operator which gets our resulting array q5 in the proper structure required for use of Markov.

```
See the income quintile for Colorado and Massachusetts
q5[4, :]
q5[19, :]
```

The first quintile is 0 and fifth quintile is 4.

To summarize the transition dynamics for all units, you instantiate a Markov object:

```
m5 = pysal.Markov(q5)
m5.transitions
```

Assuming you can treat these transitions as a first order Markov chain, you can estimate the transition probabilities:

```
m5.p
```

as well as the long run steady state distribution:

```
m5.steady_state
```

With the transition probability matrix in hand, you can estimate the first mean passage time:

```
pysal.ergodic.fmpt(m5.p)
```

Thus, for a state with income in the first quintile, it takes on average 12 years for it to first enter the second quintile, 31 to get to the third quintile, 52 years to enter the fourth, and 121 years to reach the richest quintile.

Thus far you have treated all the spatial units as independent to estimate the transition probabilities. This hides a number of implicit assumptions. First, the transition dynamics are assumed to hold for all units and for all time periods. Second, interactions between the transitions of individual units are ignored. In other words regional context may be important to understand regional income dynamics, but the classic Markov approach is silent on this issue.

PySAL includes a number of spatially explicit extensions to the Markov framework. The first is the spatial Markov class that you illustrate here. You first are going to transform the income series to relative incomes (by standardizing by each period by the mean):

```
tpci = pci.transpose()
rpci = tpci / (tpci.mean(axis = 0))
```

Next, you require a spatial weights object, and here you will create one from an external GAL file:

```
w = pysal.open("../pysal/examples/state48_percapinc_queen.gal").read()
w.transform = 'r'
```

Finally, you create an instance of the Spatial Markov class using 5 states for the chain:

```
sm = pysal.Spatial_Markov(rpci, w, fixed = True, k = 5)
```

Here you are keeping the quintiles fixed, meaning the data are pooled over space and time and the quintiles calculated for the pooled data. This is why you first transformed the data to relative incomes. You can next examine the global transition probability matrix for relative incomes:

Here you are keeping the quintiles fixed, meaning the data are pooled over space and time and the quintiles calculated for the pooled data. This is why you first transformed the data to relative incomes. You can next examine the global transition probability matrix for relative incomes:

```
sm.p
```

The Spatial Markov allows us to compare the global transition dynamics to those conditioned on regional context. More specifically, the transition dynamics are split across economies who have spatial lags in different quintiles at the beginning of the year. In our example you have 5 classes, so 5 different conditioned transition probability matrices are estimated:

```
for p in sm.P:
        print p
```

(Make sure you have an indentation before print p)

The probability that a 1st quintile state remains 1st quintile is 0.965 if it is near 1st quintile states but this drops to 0.81 if it is near 2nd quintile states and 0.4 if it is near highest quintile states.

You can also explore the different steady state distributions implied by these different transition probabilities:

```
sm.S
```

The long run distribution for states with poor (rich) neighbors has 0.0 (0.02) of the values in the first quintile, 0.0 (0.23) in the second quintile, 0.0 (0.27) in the third, etc. And, finally the first mean passage times:

```
for f in sm.F:
        print f
```

States with incomes in the second quintile with neighbors in the first quintile return to the first quintile after 13 years, after leaving the first quintile. They enter the fourth quintile 45 years after leaving the first quintile, on average.

The Spatial Markov conditions the transitions on the value of the spatial lag for an observation at the beginning of the transition period. An alternative approach to spatial dynamics is to consider the joint transitions of an observation and its spatial lag in the distribution. By exploiting the form of the static LISA and embedding it in a dynamic context you develop the LISA Markov in which the states of the chain are defined as the four quadrants in the Moran scatter plot.

```
pci = np.array([f.by_col[str(y)] for y in range(1929,2007)])
trpci = np.array([f.by_col[str(y)] for y in range(1929, 2007)]).transpose()

w = pysal.open("../pysal/examples/state48_percapinc_queen.gal").read()

lm = pysal.LISA_Markov(trpci, w)
lm.classes
lm.transitions
```

and the estimated transition probability matrix is:

```
lm.p
```

Interpret the resulting matrix as follows. Thus in the diagram shown below the probability that a HH state will stay HH in the next time period is 0.929 and a LL state will stay LL is 0.947 and LH state will become HH is 0.079.

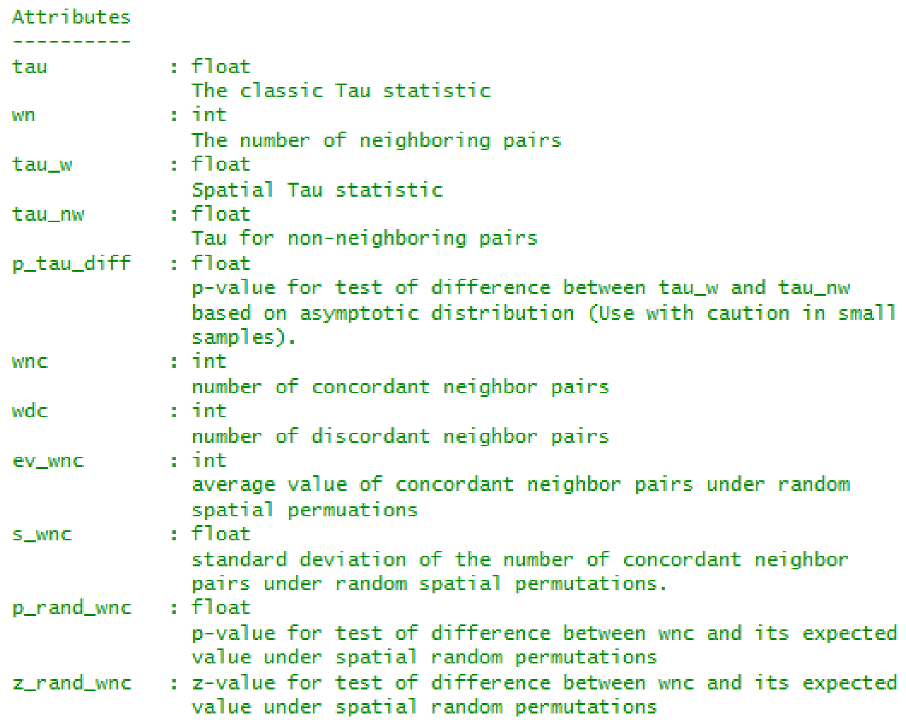
	t+1			
t	HH	LH	LL	HL
HH	0.929	0.037	0.004	0.031
LH	0.079	0.859	0.060	0.002
LL	0.005	0.020	0.947	0.028
HL	0.053	0.002	0.070	0.875

Finally the first mean passage time for the LISAs is:

```
pysal.ergodic.fmpt(lm.p)
```

Kendall's τ is based on a comparison of the number of pairs of n observations that have concordant ranks between two variables. For spatial dynamics in PySAL, the two variables in question are the values of an attribute measured at two points in time over n spatial units. This classic measure of rank correlation indicates how much relative stability there has been in the map pattern over the two periods. The spatial τ decomposes these pairs into those that are spatial neighbors and those that are not, and examines whether the rank correlation is different between the two sets.

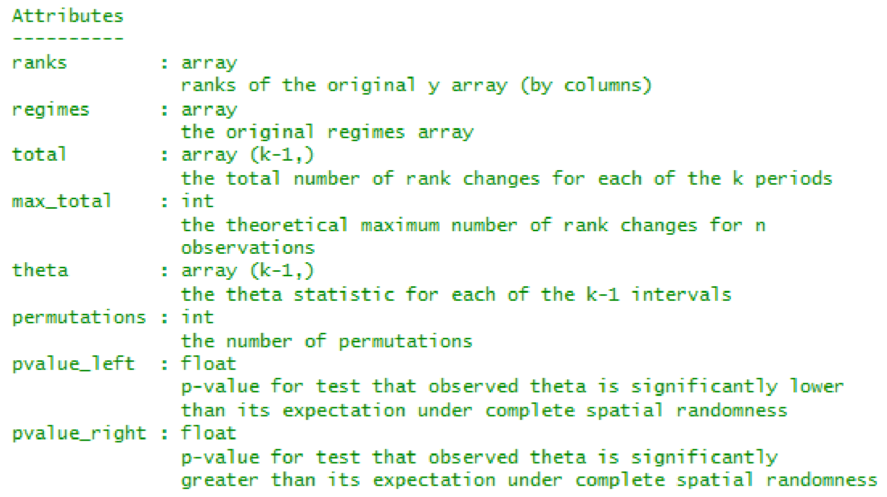
```
regime = np.array(f.by_col['SRegion'])
w = pysal.weights.regime_weights(regime)
y=trpci
np.random.seed(10)
res = [pysal.SpatialTau(y[:, i], y[:, i+1], w, 99) for i in range(6)]
for r in res:
        "%8.3f %8.3f %8.3f"%(r.wnc, r.ev_wnc, r.p_rand_wnc)
...
```



```
Attributes
----------
tau          : float
               The classic Tau statistic
wn           : int
               The number of neighboring pairs
tau_w        : float
               Spatial Tau statistic
tau_nw       : float
               Tau for non-neighboring pairs
p_tau_diff   : float
               p-value for test of difference between tau_w and tau_nw
               based on asymptotic distribution (Use with caution in small
               samples).
wnc          : int
               number of concordant neighbor pairs
wdc          : int
               number of discordant neighbor pairs
ev_wnc       : int
               average value of concordant neighbor pairs under random
               spatial permuations
s_wnc        : float
               standard deviation of the number of concordant neighbor
               pairs under random spatial permutations.
p_rand_wnc   : float
               p-value for test of difference between wnc and its expected
               value under spatial random permutations
z_rand_wnc   : z-value for test of difference between wnc and its expected
               value under spatial random permutations
```

For a sequence of time periods, θ measures the extent to which rank changes for a variable measured over n locations are in the same direction within mutually exclusive and exhaustive partitions (regimes) of the n locations. Theta is defined as the sum of the absolute sum of rank changes within the regimes over the sum of all absolute rank changes

```
np.random.seed(10)
t = pysal.Theta(y, regime, 999)
t.theta
t.pvalue_left
```



```
Attributes
----------
ranks        : array
               ranks of the original y array (by columns)
regimes      : array
               the original regimes array
total        : array (k-1,)
               the total number of rank changes for each of the k periods
max_total    : int
               the theoretical maximum number of rank changes for n
               observations
theta        : array (k-1,)
               the theta statistic for each of the k-1 intervals
permutations : int
               the number of permutations
pvalue_left  : float
               p-value for test that observed theta is significantly lower
               than its expectation under complete spatial randomness
pvalue_right : float
               p-value for test that observed theta is significantly
               greater than its expectation under complete spatial randomness
```

Time space Interaction tests are also available in Pysal. This class requires the input of a point shapefile. The shapefile must contain a column that includes a timestamp for each point in the dataset. The class requires that the user input a path to an appropriate shapefile and the name of the column containing the timestamp. In this example, the appropriate column name is ‘T’. You start by importing Numpy, PySAL and the interaction module:

```
import numpy as np
import pysal
import pysal.spatial_dynamics.interaction as interaction
np.random.seed(100)

path = "../pysal/examples/burkitt"
events = interaction.SpaceTimeEvents(path,'T')
```

Next, you run the Knox test with distance and time thresholds of 20 and 5, respectively. This counts the events that are closer than 20 units in space, and 5 units in time.

```
result = interaction.knox(events,delta=20,tau=5,permutations=99)
```

Finally you examine the results. You call the statistic from the results dictionary. This reports that there are 13 events close in both space and time, based on our threshold definitions.

```
print(result['stat'])
```

Then you look at the pseudo-significance of this value, calculated by permuting the timestamps and rerunning the statistics. Here, 99 permutations were used, but an alternative number can be specified by the user. In this case, the results indicate that you fail to reject the null hypothesis of no space-time interaction using an alpha value of 0.05.

```
print("%2.2f"%result['pvalue'])
```

Next, you run the modified Knox test with distance and time thresholds of 20 and 5,respectively. This counts the events that are closer than 20 units in space, and 5 units in time.

```
result = interaction.modified_knox(events,delta=20,tau=5,permutations=99)
```

Finally you examine the results. You call the statistic from the results dictionary. This reports a statistic value of 2.810160.

```
print("%2.8f"%result['stat'])
2.81016043
```

Next you look at the pseudo-significance of this value, calculated by permuting the timestamps and rerunning the statistics. Here, 99 permutations were used, but an alternative number can be specified by the user. In this case, the results indicate that you fail to reject the null hypothesis of no space-time interaction using an alpha value of 0.05.

```
print("%2.2f"%result['pvalue'])
```