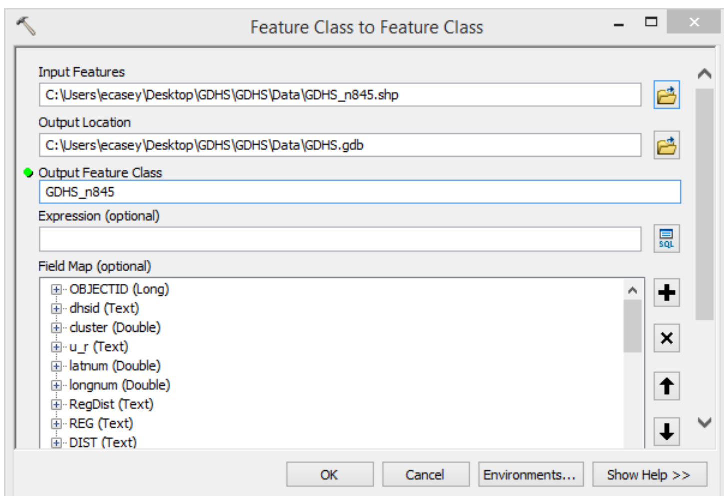
Title: “Document for preparing and conducting LTM-ESF analysis”
Author: Evan D. Casey
Date: September 2015
Note: Instructions derived from “Steps for creating and using eigenvectors for space time analysis” author Li An

Required Data: GDHS_n845.shp (in GDHS zip file)

**Step 1: Setup**
Unzip the GDHS zip file to an adequate location such as the C:\ directory.

Launch the ArcMap software. Using the ArcMap Catalog window, locate the unzipped GDHS (C:\GDHS) directory--if the directory path is not visible in the Catalog window use the Connect to Folder tool to locate and make available the directory. Inside the GDHS directory is a folder named Data (C:\GDHS\Data); click the plus sign next to the GDHS folder to ‘expand’ and right-click the Data folder. From the dropdown menu select New > File Geodatabase. Name the resulting geodatabase GDHS.

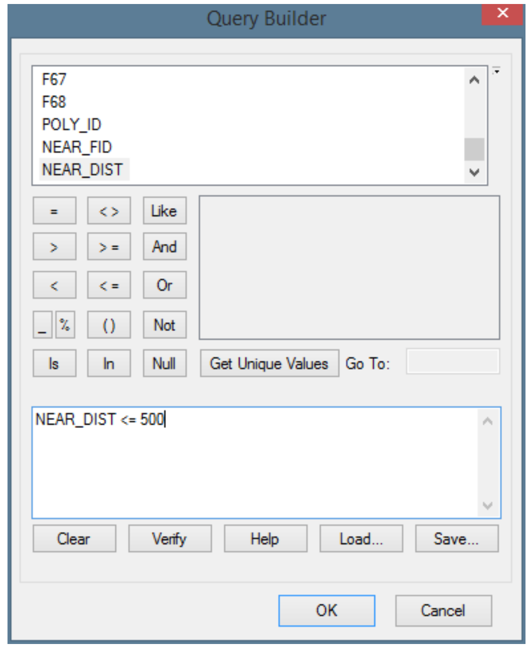
**Step 2: Add Dataset**
Expand the Data folder so as to make the newly created geodatabase visible in the Catalog window. Right-click the GDHS.gdb (C:\GDHS\Data\GDHS.gdb), selecting Import > Feature Class (Single) from the dropdown menu. The Feature Class to Feature Class wizard will appear. Click the open-folder button located next to the Input Features text-box. From the pop-up window navigate to and select the GDHS_n845 shapefile (C:\GDHS\Data\GDHS_n845.shp), then click Add. The shapefile’s directory path will then appear in the Input Features text-box. Click inside the Output Feature Class text-box and type in “GDHS_n845” (without quotation marks) to name feature class. When finished, click the OK button at the bottom of the wizard.




The GDHS_n845 feature class should then be automatically added to the ArcMap data frame and Table of Contents, but, if not, drag and drop the feature class from its location in the Catalog window (C:\GDHS\Data\GDHS.gdb\GDHS_n845) onto the data frame.

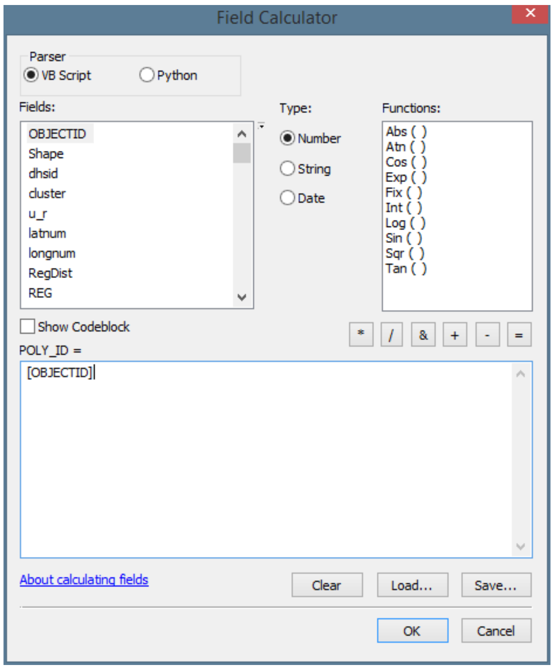
**Step 3: Data Select**
Next, open the Select tool (Toolboxes > System Toolboxes > Analysis Tools > Extract > Select) which will be used to make a new feature class containing a subset of the GDHS_n845 data. In the Select wizard, set the Input Feature to the GDHS_n845 feature class by selecting it form the dropdown list. Name the Output Feature Class u199 and locate it in the GDHS geodatabase so that its path (C:\GDHS\Data\GDHS.gdb\u199) appears in the text-box. Enter “NEAR_DIST <= 500” (without quotation marks) into the

Expression text-box—telling the tool to select data points who's nearest neighbor is 500 meter or less in distance. When done, click the OK button at the bottom of the wizard.



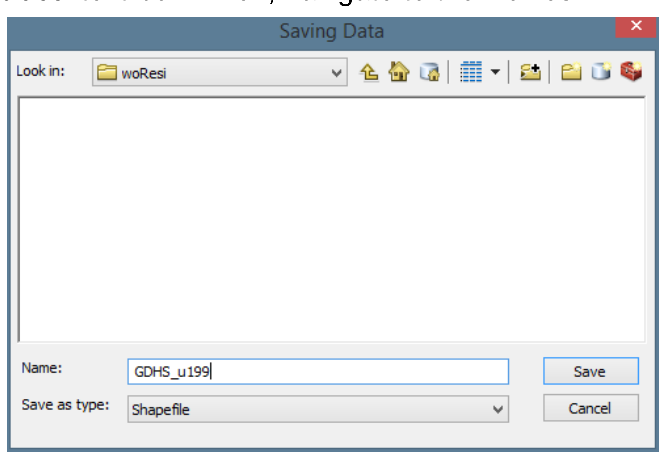

The new feature class, u199, will be added to the GDHS geodatabase as well as the ArcMap data frame and Table of Contents.

Right-click the u199 feature class in the Table of Contents window and select Open Attribute Table from the dropdown menu. Drag the horizontal scroll bar to the far right so as to make the POLY_ID attribute column visible. Right-click the POLY_ID column heading and select Field Calculator… from the dropdown menu. Click the Yes button to dismiss the warning if the intermediate window appears. With the Field Calculator Window now visible double-click OBJECTID from the Fields list so that [OBJECTID] appears in the lower text-box; then, click the OK button to reassign the POLY_ID values.

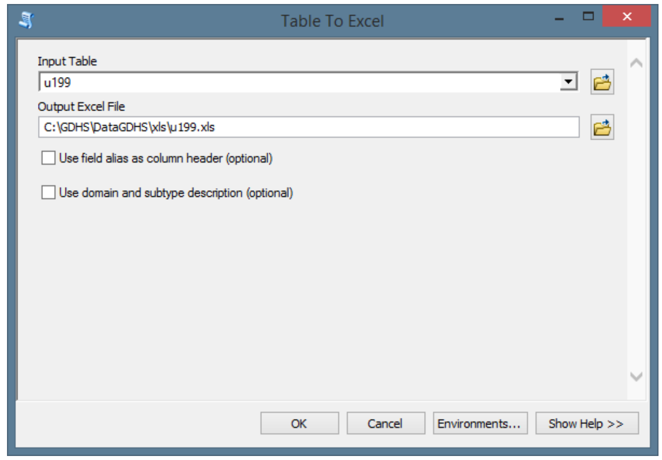



**Step 4: Export Shapefile**

Right-click the u199 feature class in the Table of Contents window and select Data > Export Data… from the dropdown menu. In the Export Data wizard, click the Browse button adjacent to the 'Output feature class' text-box. Then, navigate to the woResi folder (C:\GDHS\Shapefiles\woResi) so that its name appears at the top of the window in the 'Look in' text-box. At the bottom of the window, in the 'Name' text-box type "GDHS_u199" (without quotation marks) and make sure 'Shapefile' is selected as the 'Save as type', just below. When complete click the Save button then the OK button. When asked about adding the data as a map layer decline by clicking the No button.



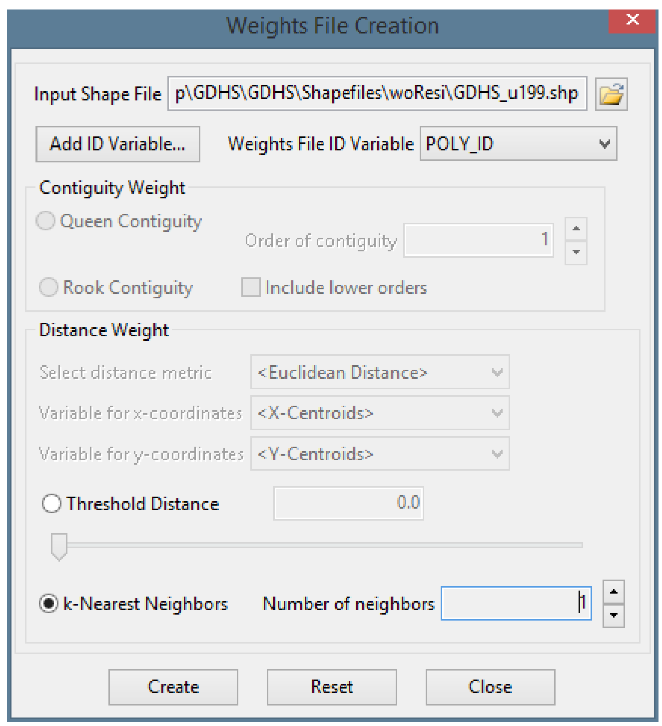
**Step 5: Export Table**
Launch the Table to Excel tool (Toolboxes > System Toolboxes > Conversion Tools > Excel > Table to Excel). For the Input Table select the u199 feature class. Name the Output Excel File “u199” (without quotation marks) and save it in the xls sub-directory of the DataGDHS file so that the file’s path (C:\GDHS\DataGDHS\xls\u199.xls) appears in the text-box. Click the OK button, when finished.

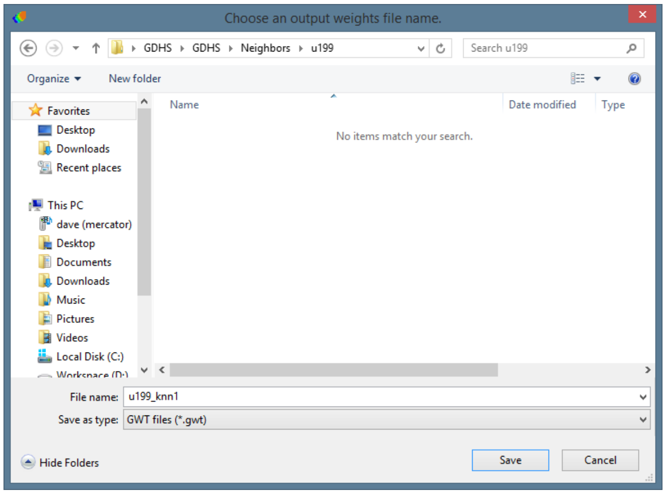



Open the u199.xls using the Microsoft Excel software. Save a copy of the file in the DataGDHS folder (C:\GDHS\DataGDHS) as an *.xlsx formatted file.

[Here, one of good places to repeat process selecting different range if want to run analysis on multiple datasets]

**Step 6: Create Weights**
Launch the GeoDa software. Open the GDHS_u199 which was created earlier by clicking File > New Project From > ESRI Shapefile (*.shp) and selecting the file from its save location (C:\GDHS\Shapefiles\woResi\GDHS_u199.shp). Then, select Tools > Weights > Create bringing up the Weights File Creation wizard. From the Weights File ID Variable dropdown list select POLY_ID. Further options will then become available. Select the k-Nearest Neighbors option near the bottom of the wizard and change the Number of Neighbors to a value of 1. Click the Create button at the bottom of the wizard. Navigate to the Neighbors folder (C:\GDHS\Neighbors) and click the Create New Folder button and name the new folder “u199” (without quotation marks). Open the newly created u199 folder (C:\GDHS\Neighbors\u199). Near the bottom of the window, type in the ‘File name’ text-box “u199_knn1” (without quotation marks) and then click the Save button.

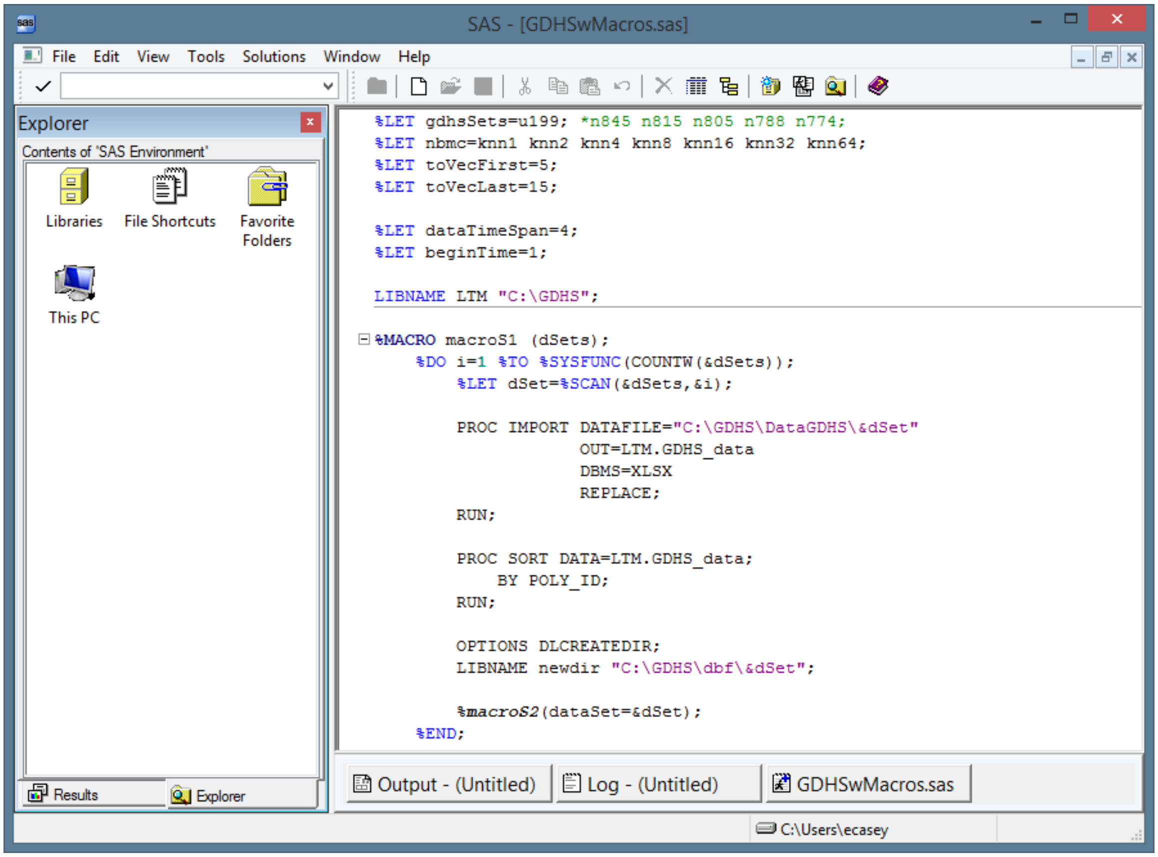





Back at Weights File Creation wizard, change the Number of neighbors value to 2 and click the Create button. In the Neighbors/u199 folder save the new weights file as “u199_knn2” (without quotation marks). Repeat this process, changing the Number of neighbors value each time, until a weights file is created for 1, 2, 4, 8, 16, 32 and 64 k-nearest neighbors.

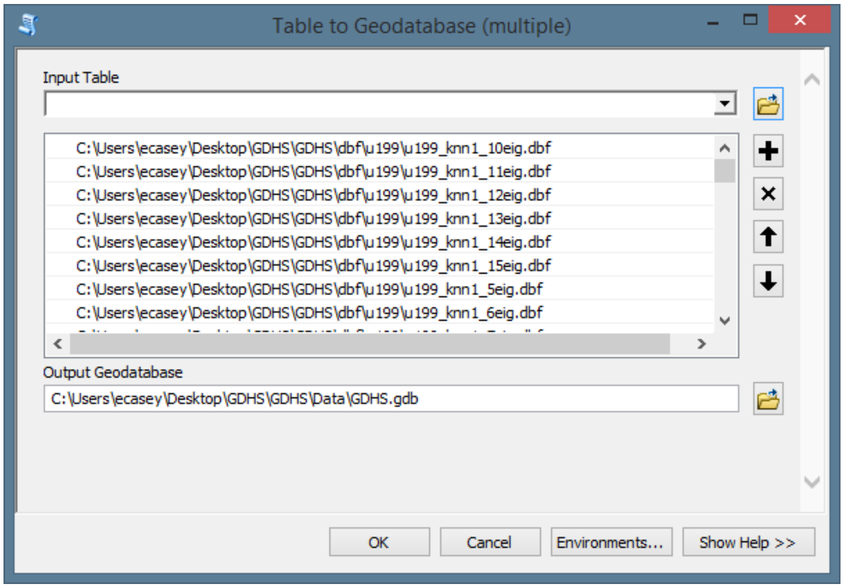
**Step 7: Generate Neighbor Matrix**
Launch the R GUI statistical software. Click the Open Script button (or by navigating File > Open Script…) and open the script NeighborMatrix.r (C:\GDHS\Scripts\NeighborMatrix.r). A window will appear with the script’s code visible. At the top of the script the Set Working Directory method is visible ( setwd(“C:/GDHS”) ); if the unzipped GDHS directory was saved someplace other than in the C:\ directory, alter the setwd contents to reflect this change. Next, look down a few lines to where the shapeList variable is defined; there, to the right, is a list of dataset names enclosed in quotation marks and separated by commas [e.g. c(“u199”, “u199_s421”, “u199_s421_r225”)]. Modify this list, if necessary, to contain the names of the datasets which were created in previous steps and are to be used in further analysis [e.g. c(“u199”)]. When done making any necessary changes to the script click Edit > Run All. After the script has finished executing, csv neighbor matrixes will be available in the dataset’s corresponding Neighbors directory folder (e.g. C:\GDHS\Neighbors\u199).

**Step 8: Run Model**
Launch the SAS statistical software. Open the SAS script GDHSwMacros.sas (C:\GDHS\Scripts\GDHSwMacros.sas). Similar to the previously run R script, the datasets to be processed by SAS can be changed within the code; to do this, type in the name(s) of desired dataset(s) to the right of the gdhsSets variable, located at the top of the script. The dataset names should be without containing quotation marks and separated from one another by a space (e.g. %LET gdhsSets=u199 u199_s421 u199_s421_r225;). The range of eigenvectors to be analyzed in the iterative process may also be changed by setting the toVecFirst and toVecLast variables. Lastly, make sure the LIBNAME library path matches that of the unzipped GDHS directory (C:\GDHS). Run the script in its entirety when finished making any necessary alterations to the script. When the script is finished executing there will have been created a *.dbf file for each combination of dataset, k-nearest neighbors and number of eigenvectors used. The *.dbf files will have been saved to the dbf folder (C:\GDHS\dbf) and in a dataset-specific sub-directory (e.g. C:\GDHS\dbf\u199).




**Step 9: Import dbf**

Once again, launch the ArcMap software. Right-click the GDHS geodatabase (C:\GDHS\Data\GDHS.gdb) and select Import > Table (multiple)… from the dropdown menu. The Table to Geodatabase (multiple) wizard will open. Click the Open Folder button to the right of the Input Table text-box. Navigate to the dbf sub-directory of the desired dataset (e.g. C:\GDHS\dbf\u199). Select all of the folder's contents and click the Add button. Returning to the Table to Geodatabase (multiple) wizard click the OK button. ArcMap will then import each of the *.dbf files into the GDHS geodatabase.




Next, open the python scripting window in ArcMap by selecting Geoprocessing > Python. Then, open the python script GDHS_dbfJoin.py (C:\GDHS\Scripts\ GDHS_dbfJoin.py). Copy each line (seen below) from the script to the ArcMap python window and execute each individually, being sure that the ‘env.workspace’ location matches that of the GDHS.gdb (C:\GDHS\Data\GDHS.gdb) and the ‘n’ contains the desired dataset(s)—analogous to the R and SAS steps.

---

```
import arcpy
from arcpy import env

env.workspace = "C:\GDHS\Data\GDHS.gdb"

n = ["u199","u199_s421","u199_s421_r225"]
knn = [1,2,4,8,16,32,64]
```

---

Afterwards, copy into the remaining python code (a for-loop) into the ArcMap python window and execute the script by pressing the Enter key twice. The script will add the SAS *.dbf output to the corresponding shapefile attribute table.

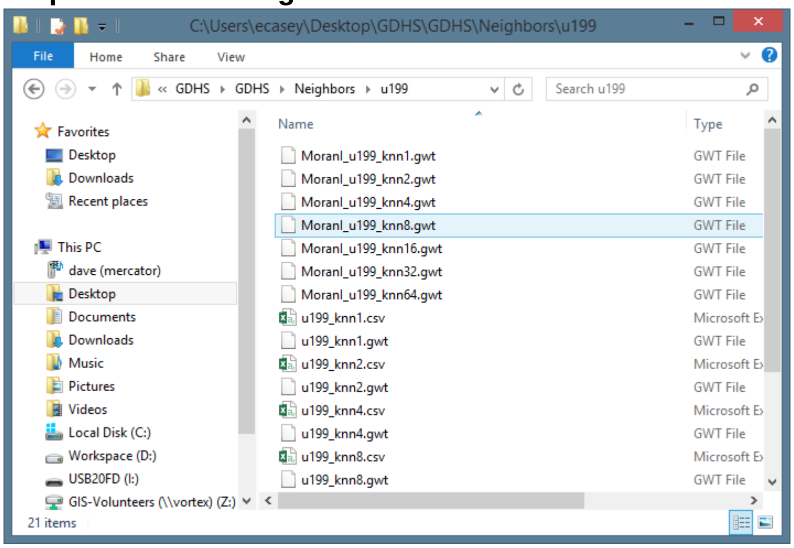
---

```
for i in n:
    inFeature = "%s" % (i)
    for j in knn:
        for k in xrange(5,16):
            fieldName = "knn%d_e%d" % (j,k)
            arcpy.AddField_management(inFeature,fieldName,"FLOAT")
            joinTable = "%s_knn%d_%deig" % (i,j,k)
            arcpy.AddJoin_management(inFeature,"POLY_ID",joinTable,"POLY_ID")
            fieldName = "%s.knn%d_e%d" % (i,j,k)
            expression = "!%s_knn%d_%deig.resi1!" % (i,j,k)
            arcpy.CalculateField_management(inFeature,fieldName,expression,"PYTHON_9.3")
            arcpy.RemoveJoin_management(inFeature,joinTable)
            print "%s populated" % (fieldName)
```

---

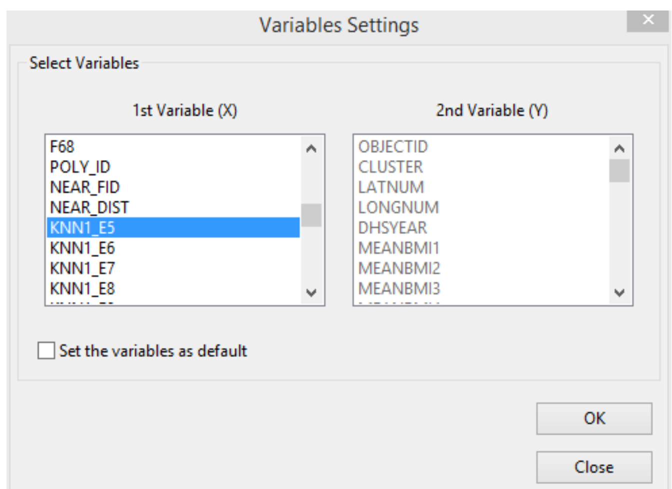
**Step 10: Export Shapefile**
Repeat step 4, this time adding "_resi" (without quotation marks) to the end of the shapefile name, and saving to the wResi folder (C:\GDHS\Shapefiles\wResi).

**Step 11: Create Weights**




Repeat step 6, opening instead the wResi shapefile (C:\GDHS\Shapefiles\wResi\GDHS_u199_resi.shp) and saving the resulting weight files with prefix "MoranI_" (without quotation marks).

**Step 12: Calculate Z-Scores**
With the shapefile still open in GeoDa select Space > Univariate Moran's I from the menu bar. The Variables Settings window will open. From the '1st Variable (X)' list select variable knn1_e5 so it is highlighted, then click OK. The Select Weight window will then appear. Click the open folder button to the right of the 'Select from file (gal, gwt)' text box. Navigate to the u199 Neighbors directory (C:\GDHS\Neighbors\u199) and select the MoranI_u199_knn1.gwt file, then click Open. With the file's path name now filling the text box click the OK button. A window will open named Moran Scatter Plot; click right-click inside the window and select Randomization > 999 Permutations from the dropdown menu. In the resulting Randomization window the Moran's I and z-score are displayed.




Repeat the process selecting the next variable (e.g. knn1_e6) being sure that the MoranI weights file matches the neighbors of the variable (e.g. knn1, knn2, etc.).