Spatial Analysis – Raster data analysis

Lecture 4-5
2/11-18/2007
Recap - raster data

- ESRI grid, image format, geodatabase raster dataset and raster catalog
- Column and row
- Cell
- Cell value
- Cell size
- Mixed pixels
- Raster data structure
  - Cell by cell (DEM and satellite image)
    - BSQ (band-interleaved), easy to retrieve spatial information
    - BIP (pixel-interleaved), easy to retrieve spectral information
    - BIL (row-interleaved), fairly for both spatial and spectral
  - Run-length encoding
  - Quad-tree
  - Wavelet (MrSID, ECW, JPEG2000)
# Raster Data Encoding: Mixed Pixel Problems

<table>
<thead>
<tr>
<th>Water dominates</th>
<th>Winner takes all</th>
<th>Edges separate</th>
</tr>
</thead>
<tbody>
<tr>
<td>W W G</td>
<td>W G G</td>
<td>W E G</td>
</tr>
<tr>
<td>W W W G</td>
<td>W W G</td>
<td>W E G</td>
</tr>
<tr>
<td>W W W W G</td>
<td>W G G</td>
<td>E E G</td>
</tr>
</tbody>
</table>

Spatial Analysis: Dr. Feng Qiu
Details

Spatial Analyst Tools
- Conditional
- Density
- Distance
- Extraction
- Generalization
- Groundwater
- Hydrology
- Interpolation
- Local
- Map Algebra
- Math
- Multivariate
- Neighborhood
- Overlay
- Raster Creation
- Reclass
- Surface
- Zonal
Spatial Analyst extension

- Most of the functions in the Spatial Analyst Tools are also integrated into the Spatial Analyst extension.
Types of spatial analysis (1)

- Mapping distance
  - Straight line (or Euclidean) distance
  - Cost Weighted Distance
- Mapping density
- Interpolating to raster
  - Inverse Distance Weighted
  - Spline
  - Kriging
- Performing surface analysis
  - Contour
  - Slop
  - Aspect
  - Hillshade
  - Viewshed
  - cut/fill
Types of spatial analysis (2)

- Statistics
  - Cell statistics
  - Neighborhood statistics
  - Zonal statistics
- Reclassification
- Raster Calculator (map algebra)
- Conversion of vector and raster
1. Mapping distance

- **Straight Line Distance** function measures the straight line distance from each cell to the closest source.

- **Allocation function** allows you to identify which cells belong to which source based on straight line distance.

- The **Cost Weighted Distance** function modifies the Straight Line Distance by some other factor, which is a cost to travel through any given cell. For example, it may be shorter to climb over the mountain to the destination, but it is faster to walk around it. Cost can be money, time, or preference.

- The Distance and Direction raster datasets are normally created from **Cost Weighted Distance** function to serve as inputs to the pathfinding function, the *shortest (or least-cost) path*. 
Global Operations

- Global operations
  - Operations that compute an output grid where the value of each output cell is potentially a function of all the cells in the input grid.
  - Correspond to Tomlin’s focal operations in extended vicinities, considered to the extreme case of focal operation where the neighborhood is potentially entire input grid.

- Example of operations
  - Global statistics and Data slicing
  - Euclidean distance analysis
    - FindDistance, EucDistance, Assign Proximity, EucAllocation
  - Weighted distance measurement
    - CostDistance
  - Least-cost path determination
    - CostPath
**Global Statistical Examples**

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td></td>
</tr>
</tbody>
</table>

- Global Maximum: 9
- Global Minimum: 1
- Global Range: 8
- Global Majority: 5

Input Grid
Straight line/Euclidean Distance

- **Straight Line/Euclidean Distance**
  - Measure the straight line distance from every cell to the nearest cell containing an object of interest (source).
  - The distance is measured from cell center to cell center using the Pythagorean theorem.

- **Source**
  - Vector features (point, line, polygon)
  - Valid grid cells (Non-null cells)

- **Output grids**
  - Continuous distance grid
  - Direction and allocation grids (Optional)
Euclidean Distance Based Analysis

- **Straight Line/Euclidean Distance**
  - Measure the straight line distance from every cell to the nearest cell containing an object of interest (source).
  - The distance is measured from cell center to cell center using the Pythagorean theorem.
- **Source**
  - Vector features (point, line, polygon)
  - Valid grid cells (Non-null cells)
- **Output grids**
  - Continuous distance grid
  - Direction and allocation grids (Optional)
The straight line distance to the nearest town from every location.
Proximity / Euclidean Allocation

- **Assign Proximity/Euclidean Allocation**
  - Allocate each cell to the nearest source and assign it the ID/value of the nearest source.
  - The nearest source is determined by the Euclidean distance.
  - It is the GRID equivalent of Thiessen polygon and a way of assigning service area.

**Source Grid**

**Output Grid**

**Euclidean Allocation**
Why use the Allocation function?

Use the Allocation function to perform analyses such as:

- Identifying the customers served by a series of stores
- Finding out which hospital is the closest
- Finding areas with a shortage of fire hydrants
- Locating areas that are not served by a chain of supermarkets

- Allows you to identify which cells belong to which source based on straight line distance function or cost weighted distance function.

1, 2 are shopping centers
Cost weighted distance

- Find the least accumulative cost from each cell to the nearest, cheapest source. Cost can be money, time, or preference.

The functions that perform cost weighted distance mapping are similar to the Straight Line Distance functions, but instead of calculating the actual distance from one point to another, they compute the accumulative cost of traveling from each cell to the nearest source, based on the cell’s distance from each source and the cost to travel through it—for example, it is easier to walk through a meadow than a swamp.
Why use the Cost Weighted Distance function?

Cost weighted distance modeling is useful whenever movement is based on geographic factors, such as animal migration studies or consumer travel behavior. Cost weighted distance may also be used to minimize construction costs for routing new roads, transmission lines, or pipelines.

The straight line distance between two points is not necessarily the best. In the graphic to the left, the shortest path over the mountain takes three hours. The longer path around only takes two hours. If time were a cost, then the route with the longer distance should be taken. However, the aim may be to climb over the mountain. Applying cost weighted distance enables you to specify preferences in your input data. It may, for example, take longer to travel over the mountain due to steep slopes, so steep slopes should be given a higher cost when finding a suitable path from A to B.
Weighting datasets according to percent influence

The next step in producing the cost raster is to add the reclassified datasets together. The simplest approach is to just add them together. However, you may know that some factors are more important than others. For instance, avoiding steep slopes may be twice as important as the landuse type, so you might, for example, give this dataset an influence of 66 percent and the landuse dataset an influence of 34 percent, to make 100 percent. The following diagram shows the conceptual process:

```
<table>
<thead>
<tr>
<th>3</th>
<th>2</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>3</td>
</tr>
</tbody>
</table>
Slope

* 0.66

<table>
<thead>
<tr>
<th>1.98</th>
<th>1.32</th>
<th>1.32</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.64</td>
<td>2.64</td>
<td>1.32</td>
</tr>
<tr>
<td>1.98</td>
<td>3.96</td>
<td>1.98</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>
Landuse

* 0.34

<table>
<thead>
<tr>
<th>0.34</th>
<th>0.34</th>
<th>0.34</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.04</td>
<td>2.04</td>
<td>0.34</td>
</tr>
<tr>
<td>2.04</td>
<td>1.02</td>
<td>1.02</td>
</tr>
</tbody>
</table>

1.98 + 2.64 + 1.32 + 1.98 + 2.04 + 2.04 + 1.02 + 1.02 = 10.94
```
**Direction**

The cost-weighted distance raster tells you the least accumulated cost of getting from each cell to the nearest source, but it doesn’t tell you which way to go to get there. The direction raster provides a road map, identifying the route to take from any cell, along the least-cost path, back to the nearest source.

![Image of cost weighted, direction, and direction coding rasters]

The algorithm for computing the direction raster assigns a code to each cell that identifies which one of its neighboring cells is on the least-cost path back to the nearest source. In the direction coding diagram above, 0 represents every cell in the cost-weighted distance raster. Each cell is assigned a value representing the direction of the nearest, cheapest cell on the route of the least costly path to the nearest source.
Shortest path

- The shortest path function determines the path from a destination point to a source. Once you have performed the cost weighted distance function, creating distance and direction raster, you can then computer the least-cost or shortest path from a chosen destination to your source point.

The **purple** line represents a cost distance where each input raster (landuse and slope) had the same influence. The **red** line represents a cost distance where the slope input raster had a weight (or influence) of 66 percent.
2. Mapping density

- Spread point values over a surface

The graph gives an example of density surface. when added together, the population values of all the cells equal to the sum of the population of the original point layer.
3. spatial interpolation

- Will be covered in the geostatistic lectures
4. Surface analysis: Contours

- Contours are polylines that connect points of equal value, such as elevation, temperature, precipitation, pollution, or atmospheric pressure.
Slope

Degree of slope = $\theta$

Percent of slope = \( \frac{\text{rise}}{\text{run}} \times 100 \)

\[
\tan \theta = \frac{\text{rise}}{\text{run}}
\]

Degree of slope = 30  
Percent of slope = 58

Degree of slope = 45  
Percent of slope = 100

Degree of slope = 76  
Percent of slope = 375

Output slope dataset
Aspect

It is measured clockwise in degrees from 0—due north—to 360—again due north, coming full circle. The value of each cell in an aspect dataset indicates the direction the cell’s slope faces. Flat slopes have no direction and are given a value of -1.
Hillshade

Setting a hypothetical light source and calculating the illumination values for each cell in relation to neighboring cells. It can greatly enhance the visualization of a surface for analysis or graphical display.

Azimuth is the angular direction of the sun, measured from north in clockwise degrees from 0 to 360. An azimuth of 90 is east. The default is 315 NW.

Altitude is the slope or angle of the illumination source above the horizon. The units are in degrees, from 0—on the horizon—to 90 degrees—overhead. The default is 45 degrees.

Azimuth 315°, altitude 45°
Viewshed

- Viewshed identifies the cells in an input raster that can be seen from one or more observation points or lines.
- It is useful for finding the visibility. For instance, finding a well-exposed places for communication towers.
5. Statistics: Cell statistics (local function)

- A statistic for each cell in an output raster is based on the values of each cell of multiple input rasters.
  - For instance, to analyze the average crop yield over a 10-year period.
- Majority, maximum, mean, median, minimum, minority, range, standard deviation, sum, variety.
If any of the input is NODATA, the output is NODATA

If all the inputs are integer, the output is integer. If any of the inputs are floating point, the output is floating point.

Expression: \( \text{MAX}(\text{INGRID1}, \text{INGRID2}, \text{INGRID3}) \)
Expression: \( \text{MEAN}({\text{INGRID1}}, {\text{INGRID2}}, {\text{INGRID3}}) \)
Neighborhood statistics (focal)

- A statistic for each cell in an output raster is based on the values of cells within a specified neighborhood: rectangle, circle, annulus, and wedge.
- Majority, maximum, mean, median, minimum, minority, range, standard deviation, sum, variety

![Image showing an example of neighborhood statistics](image)

- **Focal Minor:** the least frequent value in neighborhood
- **Focal Major:** the most frequent value in neighborhood
- **Focal Min:** the minimum value in neighborhood
- **Focal Max:** the maximum value in neighborhood
- **Focal Sum:** the total of all values in neighborhood
- **Focal Mean:** the average of all values in neighborhood
- **Focal Variety:** the number of different values in neighborhood

**Sum of 3 x 3 cell neighborhood**

**Range = max-min**
Moving Windows

- Useful for calculating local statistical functions or edge detection
- Kernel: a set of constants applied with a function, such as 1/9 being the mean of the center cell.
- Other configurations may be used when dealing with diagonal or adjacent cells
Examples of Focal Statistic Operations

(Use a 3 x 3 window)

Test

Minimum

Maximum

Mean

Variety
Zonal statistics

- computer statistics for each zone of a zone dataset based on the information in a value raster. zone dataset can be feature or raster, the value raster must be a raster.
Zone

- Any unique area(s) / group of cells with the same value.
- In most GIS, the “value” for zones must be integer
- A zone may consist of noncontiguous cell or areas
- Zones can not overlap.

Spatial Analysis: Dr. Fang Qiu
Zonal Operations

- Compute a new value for each location as a function of the existing values associated with a zone containing that location.
  - Zone layer: Define the zones, their shape, values, and locations
  - Value layer: contains the input values used in calculating output for each zone for some zonal operations
  - Output layer: repeats the output value in each cell of the zone.
Zonal Statistics Example

Zone Grid

<table>
<thead>
<tr>
<th>1</th>
<th>1</th>
<th>4</th>
<th>3</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
</tbody>
</table>

Value Grid

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>9</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Zonalmean

Output Grid

<table>
<thead>
<tr>
<th>4.4</th>
<th>4.4</th>
<th>5.3</th>
<th>4.2</th>
<th>4.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4</td>
<td>4.4</td>
<td>5.3</td>
<td>4.2</td>
<td>4.2</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>5</td>
<td>4.2</td>
<td>5.3</td>
</tr>
<tr>
<td>5</td>
<td>4.4</td>
<td>5</td>
<td>4.2</td>
<td>5.3</td>
</tr>
<tr>
<td>4.4</td>
<td>4.4</td>
<td>5.3</td>
<td>5.3</td>
<td>5.3</td>
</tr>
</tbody>
</table>
Why use zonal statistics?

You might calculate the mean elevation for each forest zone or the number of accidents along each of the roads in a town. Alternatively, you might want to know how many different types of vegetation there are in each elevation zone—variety. The graphics below show an example of the inputs and outputs from the Zonal Statistics function. The variety of vegetation species per elevation zone is displayed in the output table and chart. The most variety of species occurs at elevation levels of around 2,500 meters.
6. Reclassification

What is reclassification?

Reclassifying your data simply means replacing input cell values with new output cell values.

The input data can be any supported raster format. If you add a multiband raster, the first band will be taken and used in the reclassification.

Why reclassify your data?

There are many reasons why you might want to reclassify your data. Some of the most common reasons are:

- To replace values based on new information
- To group certain values together
- To reclassify values to a common scale—for example, for use in suitability analysis or for creating a cost raster for use in the Cost Weighted Distance function
- To set specific values to NoData or to set NoData cells to a value
7. Raster calculator

What can you do with the Raster Calculator?

The Raster Calculator provides you with a powerful tool for performing multiple tasks. You can type in Map Algebra syntax to perform mathematical calculations using operators and functions, set up selection queries, or type in Spatial Analyst function syntax. Inputs can be grid datasets or raster layers, shapefiles, coverages, tables, constants, and numbers.

Mathematical operators and functions

Operators and functions evaluate the expression only for input cells that are spatially coincident with the output cell.

Operators

**Boolean operators:** And, Or, XOr, Not

**Relational operators:** ==, >, <, <=, >=, <=

**Arithmetic operators:** *, /, -, +, Log, Exp, Sin, Cos, Sqrt

Functions

There are four groups of mathematical functions: Logarithmic, Arithmetic, Trigonometric, and Powers.

- **Exponential and logarithmic**
  - Abs, Ceil, Floor, Int, Float, InNull
  - Sin, Cos, Tan, Asin, Acos, Atan
  - Sqrt, Pow
Map Algebra

- Map algebra
  - Uses math-like expressions to create new grid themes
  - Single-factor maps are treated as variables in the expression
  - The expression can be made up of variables (maps) connected with operators and functions

$$\text{AverageCost} = \frac{(\text{HisCost} + \text{MyCost})}{2}$$
Operations on Raster Layers

- **Input variables** in the form of raster/vector layers
- **Operations** are distinct and well-defined data processing activities, including **operators and functions**
- **New raster layer** (output variable) is created after applying an operations to one or more input raster/vector layers.
- **Complex analysis can be performed by a sequence of operations.**
Classification on Operations

- Tomlin (1983, 1990) defined and organized operations on raster data model as local, focal and zonal according to the spatial scope of the operations (Geographic Information System and Cartographic Modeling, Englewood Cliffs: Prentice Hall, 1990).


- Application operations are also defined to perform a specific application.
Map Algebra Operations

- **Local operation:**
  - The value at each location in the output grid depends only on the value of corresponding cell(s) at that location in the input grid(s)

- **Focal and block operation:**
  - The value at each cell in the output grid depends the value of a specified neighborhood (or block) of that cell in the input grid

- **Zonal and regional operation:**
  - The value of each cell in the output grid depends on the value of all cells in the input grid that share the same zone or region. A zone is a set of locations that have the same value and a region is a set of connected location having same value.

- **Global operation:**
  - The value of each cell in the output grid potentially depends on all cells in the input grids.
Map Algebra Operations

- Application Operations
  - Density
  - Surface generation
  - Hydrological analysis
  - Resolution altering
  - Generalization
  - Geometric transformation
  - Data Reclassification
  - Distance analysis
    - Straight line
    - Cost Weighted
    - Allocation
    - Shortest Path
  - Surface analysis
    - Slope
    - Aspect
    - Hillshade
    - Viewshed
    - Curvature
    - Contour
NODATA in Map Algebra

- **NODATA**
  - Inadequate information or absence of data to characterize value in a cell (i.e., value does not exist)
  - NODATA != 0, Default is –9999 but user can set its value
  - Computation with NoData
    - Return NoData for the location no matter what
    - Ignore the NoData and compute with the available values

General rule—If *any* input cell is NODATA, the output cell is NODATA

\[
\begin{array}{ccc}
5 & 3 & 7 & 18 \\
10 & 13 & 14 & 15 \\
13 & 11 & 11 & 6 \\
11 & 8 & 15 & 14 & 8 \\
16 & 1 & 6 & 13 & 15 \\
\end{array}
= \begin{array}{ccc}
3 & 2 & 3 & 6 & 9 \\
7 & 8 & 9 & 8 \\
7 & 3 & 8 & 2 \\
9 & 7 & 5 & 5 & 7 \\
9 & 1 & 2 & 7 & 10 \\
\end{array}
+ \begin{array}{ccc}
2 & 1 & 4 & 9 \\
8 & 3 & 5 & 5 & 7 \\
6 & 8 & 3 & 4 \\
2 & 1 & 10 & 9 & 1 \\
7 & 0 & 4 & 6 & 5 \\
\end{array}
\]

outgrd = ingrd1 + ingrd2
Map Algebra Expression Components

- **Objects**
  - Grids, shape files
  - Numbers, Strings

- **Operators**
  - + (add), /(divid), > (greater than), etc.

- **Functions/Requests**
  - Abs(), Focalmean(), Zonalgeometry(), etc

---

**Algebraic Expression**

\[
\text{DiffDem} = \text{Abs}(\text{IdwDem} - \text{SplineDem})
\]
Map Algebra Rules in Map Calculator

- Evaluate from left to right (no precedence for different operator)
  \[ [\text{Grid1}]+[\text{Grid2}]*[\text{Grid3}]=[([\text{Grid1}]+[\text{Grid2}])*[\text{Grid3}]] \]
- Use parenthesis to control the order of evaluation
  \[ [\text{Grid1}]+([\text{Grid2}]*[\text{Grid3}])] \]
- Grids are in brackets: \([\text{Elevation}]\)
- Grids can have path names (E.g. \([c:\text{data}\text{\textbackslash}grd1]\)) if not listed in the view
- Functions must be followed by (), with parameters inside
- Valid requests automatically create grid theme in the view
  - \(\text{Aspect}([\text{Elevation}])\)
- Parameters must be comma delimited and in parentheses
  - \(\text{Hillshade} ([\text{Elevation}], 315, 45, \text{nil})\)
- Function parameters may be other function (nesting)
  - \(\text{Hillshade} (\text{Aspect}([\text{Elevation}]), (35+270)/2, 45, \text{nil})\)
- Strings are delimited with the double quote character (" ")
- Only requests that create a grid theme can be used in the Raster Calculator
Local Operators

- **Arithmetic**
  
  +, -, *, /, % (Modulus, or division remainder), ^ (raise to power)

- **Relational:**
  
  =, <>, <=, >=, <, >

- **Boolean:**
  
  AND, OR, XOR, NOT

- **Parenthesis**
  
  ( )

Spatial Analysis: Dr. Fang Qiu
Local Operators

◆ Arithmetic
  ◆ + anotherGrid : Grid
  ◆ - anotherGrid : Grid
  ◆ * anotherGrid : Grid
  ◆ / anotherGrid : Grid
  ◆ % anotherGrid : Grid
  ◆ ^ anotherGrid : Grid
  ◆ Negate : Grid

◆ Relation
  ◆ < anotherGrid : Grid
  ◆ <= anotherGrid : Grid
  ◆ <> anotherGrid : Grid
  ◆ = anotherGrid : Grid
  ◆ > anotherGrid : Grid
  ◆ >= anotherGrid : Grid

◆ Bitwise
  ◆ & anotherGrid : Grid
  ◆ | anotherGrid : Grid
  ◆ ~ : Grid
  ◆ ! anotherGrid : Grid
  ◆ >>> anotherGrid : Grid
  ◆ << anotherGrid : Grid

◆ Logical
  ◆ And anotherGrid : Grid
  ◆ Or anotherGrid : Grid
  ◆ Not : Grid
  ◆ XOR anotherGrid : Grid
Logic Value in Map Algebra

◆ Logic values

◆ True: Any none zero value EXCEPT NO DATA
◆ False: Zero value

General rule—Nonzero values are logical TRUE, zero values are FALSE:

Relational operators and some functions return logical 0’s and 1’s:

Logical grids (zero and nonzero) may be used as tests in expressions:
Conditional Processing

The Con request

- Check a condition for true or false
- Parameters include a true grid and a false grid

Example:

\[ \text{Con([ingrd]}>1, 1, 2) \]

<table>
<thead>
<tr>
<th>ingrd</th>
<th>outgrd</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1</td>
<td>2 2</td>
</tr>
<tr>
<td>3 2</td>
<td>1 1</td>
</tr>
</tbody>
</table>
IsNull and SetNull

- **IsNull** tests for No Data and returns true or false (1,0)
- **SetNull** assigns No Data to non-zero cell
- Both are often used with the Con request
  - con (IsNull([Elevation]), -9999, [Elevation])
  - SetNull([aGrid]=4, aGrid)

IsNull([Ingrd])

SetNull([Ingrd]=4, [Ingrd])
Local Operators

\[
\begin{array}{c}
\begin{array}{ccc}
9 & 9 & 7 \\
9 & 8 & 5 \\
6 & 3 & 0 \\
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccc}
0 & 0 & 2 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccc}
9 & 9 & 9 \\
9 & 8 & 6 \\
6 & 3 & 0 \\
\end{array}
\end{array}
\]

= 

\[
\begin{array}{c}
\begin{array}{ccc}
9 & 9 & 7 \\
9 & 8 & 5 \\
6 & 3 & 0 \\
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccc}
0 & 0 & 2 \\
0 & 0 & 1 \\
0 & 0 & 0 \\
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccc}
N & N & 3.5 \\
N & N & 5 \\
N & N & N \\
\end{array}
\end{array}
\]

= 

\[
\begin{array}{c}
\begin{array}{ccc}
9 & 9 & 9 \\
9 & 8 & 6 \\
6 & 3 & 0 \\
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccc}
N & N & 3.5 \\
N & N & 5 \\
N & N & N \\
\end{array}
\end{array}
\begin{array}{c}
\begin{array}{ccc}
N & N & 3.5 \\
N & N & 5 \\
N & N & N \\
\end{array}
\end{array}
\]
Local Functions

- Trigonometric Functions
  - ACos : Grid
  - ACosh : Grid
  - ASin : Grid
  - ASinh : Grid
  - ATan : Grid
  - ATan2 ( yGrid ) : Grid
  - ATanh : Grid
  - Cos : Grid
  - Cosh : Grid
  - Sin : Grid
  - Sinh : Grid
  - Tan : Grid
  - Tanh : Grid

- Exponential and Logarithmic Functions
  - Exp : Grid
  - Exp10 : Grid
  - Exp2 : Grid
  - Log : Grid
  - Log10 : Grid
  - Log2 : Grid
  - Pow ( anotherGrid ) : Grid
  - Sqr : Grid
  -.Sqrt : Grid
Local Functions

- **Local Statistics**
  - `EqualTo(aGridList):Grid`
  - `GridsGreaterThan(aGridList):Grid`
  - `GridsLessThan(aGridList):Grid`
  - `LocalStats(aGridStaTypeEnum, aGridList):Grid`

- **Local Arithmetic Functions**
  - `Abs:Grid`
  - `Ceil:Grid`
  - `Float:Grid`
  - `Floor:Grid`
  - `Fmod(anotherGrid):Grid`
  - `Int:Grid`

- **Local Reclassification**
  - `Lookup(aFieldName):Grid`
  - `Reclass(aVTab, frmField, toField, outField, noData):Grid`
  - `ReclassByClassList(aFieldName, aClassList, noData):Grid`
  - `Slice(aGridSliceTypeEnum, numZones, baseZone):Grid`

- **Local Condition Checking**
  - `Con(yesGrid, noGrid):Grid`
  - `IsNull:Grid`
  - `Pick(aGridList):Grid`
  - `SetNull(anotherGrid):Grid`
Cell Statistics

- **Majority:**
  - Determines the value that occurs most often on a cell-by-cell basis between inputs

- **Maximum:**
  - Determines the maximum value on a cell-by-cell basis between inputs

- **Mean:**
  - Computes the mean of the values on a cell-by-cell basis between inputs

- **Median:**
  - Computes the median of the values on a cell-by-cell basis between inputs

- **Minimum:**
  - Determines the minimum value on a cell-by-cell basis between inputs
Cell Statistics

- **Range: Max-min**
  - Determines the range of values on a cell-by-cell basis between inputs

- **Standard Deviation:**
  - Computes the standard deviation of the values on a cell-by-cell basis between inputs

- **Sum:**
  - Computes the sum of the values on a cell-by-cell basis between inputs

- **Variety:**
  - Determines the number of unique values on a cell-by-cell basis between inputs

- **Minority:**
  - Determines the value that occurs least often on a cell-by-cell basis between inputs
Local analysis: change detection

- **Water on Both dates**
- **Raster Calculator**
  - $([\text{DateOne}] = 1) \text{ AND } ([\text{DateTwo}] = 1)$
  - $[\text{DateOne}] \times [\text{DateTwo}]$
• **Water on Either dates**
  - Raster Calculator:
    - $$([\text{DateOne}]=1) \lor ([\text{DataTwo}]=1)$$

• **Water on Only one date**
  - Raster Calculator:
    - $$((([\text{DateOne}]=1) \lor ([\text{DateTwo}]=1)) \land \neg (([\text{DateOne}]=1) \land ([\text{DateTwo}]=1)))$$
    - $$([\text{DateOne}]=1) \oplus ([\text{DateTwo}]=1)$$
- Other land uses change to water (From DateOne to DateTwo)

  - Raster Calculator:
    - 
      
  • ([DateOne]=2) AND ([DateTwo]=1)
Spatial Analysis: Dr. Fang Qiu

- Raster Calculator:
  - \([\text{DateOne}] + [\text{DateTwo}]\)

- Raster Calculator:
  - \([\text{DateOne}] \times 10 + [\text{DateTwo}]\)
Local Analysis: Mean Annual Temperature Prediction

- **Map Calculator**
  
  \[ \text{Temp C} = 12.9 - 0.003364 \times \text{Elevation}_{\text{meter}} \]
  
  \[ \text{Temp C} = 42.7 - 0.00237 \times \text{Elevation}_{\text{meter}} - 0.766 \times \text{Latitude}_{\text{degrees}} \]
Local Analysis: Water Volume Calculation

- Data Sets
  - Digital Elevation Model (DEM)
  - Lake Shape File
  - Elevation of the Lake

- Calculation
  - Volume = Length \times Width \times Height
  - Water Volume = Cell Size^2 \times Depth
  - Depth = LakeSurface - DEM
Complex operations and functions

Sometimes hundreds of operations and functions may be necessary

DEMO !!!
8. Conversion

- Feature (polygon, polyline, points) to raster
- Raster to feature (polygon, polyline, points)
Main references

- ESRI book: Using ArcGIS Spatial Analyst
- ESRI: www.ersi.com
- Dr. Fang Qi’s lecture notes