Subject: Local Spatial Statistics Function (Getis-Ord Gi* and zGi*)
Posted by jaden on Thu, 13 Apr 2006 20:06:28 GMT
View Forum Message <> Reply to Message

```
; NAME:
;       GETIS
;
; PURPOSE:
;       compute the Getis-Ord local measure of spatial
;       autocorrelation Gi* statistic and its standardized version
;       zGi* from x-y point data
;
; CATEGORY:
;       spatial statistics
;
; CALLING SEQUENCE:
;       result = GETIS(X, Y, VALUES, RADIUS)
;
; INPUTS:
;       X --> data vector or one-column array containing the x
;       coordinates (e.g. UTM easting values)
;
;       Y --> data vector or one-column array containing the y
;       coordinates (e.g. UTM norting values)
;
;       VALUES --> data vector or one-column array containing the
;       attibute value for each x-y data point of which to examine
;       autocorrelation
;
;       NOTE: X, Y, and VALUES must have the same number of elements
;
;       RADIUS --> integer or floating point value of the search
;       radius at which to look for local spatial autocorrelation
;
; NOTES:
;       X, Y, and VALUES must have the same number of elements
;
; RETURNS:
;       6 column floating point array containing:
;       [ X, Y, VALUE, Number of Neighbours, Gi*, zGi* ]
;
; REFERENCES:
;       Wulder M. & Boots B. (1998) Local spatial autocorrelation
;       characteristics of remotely sensesd imagery assessed with the
;       Getis statistic. International Journal of Remote Sensing 19:
;       2223-22231.
;
; AUTHOR/REVISIONS:
```

```
;       Jaden Langfod (jaden@uvic.ca) - April 13, 2006

FUNCTION getis, x, y, values, radius

IF (N_ELEMENTS(x) NE N_ELEMENTS(y) OR $
  N_ELEMENTS(x) NE N_ELEMENTS(values)) THEN $
  PRINT, 'Error - x, y, and value arrays must have the same dimensions'

; initialize output array

num = FLOAT(N_ELEMENTS(values))
output = FLTARR(6,num)
output[0:2,*] = [x, y, values] ; copy original data into output array
values = REFORM(output[2,*]) ; re-set values to ensure they are FLOAT

; calculate mean and variance
; (for use in standardized Gi* calculations)

mean = MEAN(values)
;s2 = VARIANCE(values)
s2 = TOTAL((values-mean)^2)/num

; begin computing the statistic for each element

missing = 0
FOR i=0, num-1 DO BEGIN

    in = WHERE ( (x-x[i])^2 + (y-y[i])^2 LE radius^2, neighbours)
    ; find points inside search radius

    w = FLTARR(num)
    w[in] = 1 ; if within radius, assign weight of one (default is 0)

    output[3,i] = neighbours ; number of neighbours
    output[4,i] = TOTAL(w*values)/TOTAL(values) ; gi* statistic

    ; standardized Gi* statistic calculations

    Wi = TOTAL(w)

    numerator = TOTAL(w*values) - (wi*mean)
    denomanator =  SQRT(s2) * ( ( Wi * (num-Wi) ) / (num-1) )^0.5
    output[5,i] = TOTAL(numerator)/denomanator ; zGi* statistic

    IF (neighbours LE 8) THEN missing = missing + 1

ENDFOR
```

```
PRINT, num, ' points processed'
IF (missing NE 0) THEN PRINT, missing, ' points with <8 neighbours'

RETURN, output

END
```