Subject: Re: Number of points within a rectangular region Posted by btt on Fri, 31 Oct 2003 18:50:21 GMT

View Forum Message <> Reply to Message

```
Isa Usman wrote:
> Hello.
>
> I'm trying to get the indices to the points that fall within a rectangular
> region. I'm aware of David Fanning's INSIDE routine that does this but
> because I have got 100,000 points its going to involve a lot of looping. So
> I decided to use the HIST ND with reverse indices but its not giving me the
> right results. What am I missing?
>
  Thanks in advance.
> Isa
  IDL > x = [0.5, -0.5, -0.5, 0.5]
  IDL > y = [0.5, 0.5, -0.5, -0.5]
>
 IDL > z = transpose([[x],[y]])
> IDL> result = hist_nd(z,min = [0,0], max = [1,1], nbins = 1, reverse_indices
> print, ri[ri[0]:ri[1]-1]
> 0123
```

You're mostly OK - but HISTOGRAM, and therefore HIST_ND, is sensitive to data typing issues. Make sure that MIN, MAX and (if you were to use it) BINSIZE are the same type as the input data.

Ben

Hi,

Subject: Re: Number of points within a rectangular region Posted by wmconnolley on Sat, 01 Nov 2003 14:58:25 GMT View Forum Message <> Reply to Message

Ben Tupper btupper@bigelow.org wrote:

- > Isa Usman wrote:
- >> I'm trying to get the indices to the points that fall within a rectangular

```
sx = size(px)
sy = size(py)
```

IF (sx[0] EQ 1) THEN NX=sx[1] ELSE message, px not a vector IF (sy[0] EQ 1) THEN NY=sy[1] ELSE message, py not a vector

>> region. I'm aware of David Fanning's INSIDE routine that does this but >> because I have got 100,000 points its going to involve a lot of looping. So >> I decided to use the HIST_ND with reverse indices but its not giving me the

px,py - points defining polygon (will be closed automatically)

; More-info: posted to comp.lang.idl-pvwave on Wed, 01 Apr 1998 12:26:38 +0200

See-also: http://www.ecse.rpi.edu/Homepages/wrf/geom/pnpoly.html for another method,

; Note: reduce test from 1e-8 to 1e-4, since we are usually in single precision. In fact, "0.1"

>> right results. What am I missing?

-W.

; Category: maths

OR

possibly better

@comm error

This is (I hope) a vectorised version of INSIDE.

FUNCTION inside, x, y, px, py, index=index

indicies of points inside (if /index is set) Author: "Bï¿1/2rd Krane" <bard.krane@fys.uio.no> Mods: wmc - make it work with x, y as vectors

; Purpose: see if point is inside polygon

; This is better than my routine "is_inside"

Input: x, y - [vector of] points

Output: vector of 1's and 0's

would do as well.

IF (NX EQ NY) THEN N = NX ELSE message, 'Incompatible dimensions'

; Close Polygon in x $tmp_px = [px, px[0]]$ $tmp_py = [py, py[0]]$; Close Polygon in y

ı = ındgen(N,/long) ip = indgen(N,/long) + 1 ; indices 0...N-1 ; indices 1...N

nn=n_elements(x)

 $X1 = tmp_px(i)\#replicate(1,nn) - replicate(1,n)\#reform([x],nn)$ $Y1 = tmp_py(i)#replicate(1,nn) - replicate(1,n)#reform([y],nn)$ $X2 = tmp_px(ip)\#replicate(1,nn) - replicate(1,n)\#reform([x],nn)$ $Y2 = tmp_py(ip)#replicate(1,nn) - replicate(1,n)#reform([y],nn)$

dp = X2*X1 + Y1*Y2; Dot-product

Subject: Re: Number of points within a rectangular region Posted by Paul Sorenson on Sat, 01 Nov 2003 19:13:40 GMT View Forum Message <> Reply to Message

Climate Modeller, British Antarctic Survey | Disclaimer: I speak for myself I'm a .signature virus! copy me into your .signature file & help me spread!

Have you taken a look at IDLanROI::ContainsPoints()? That routine might be useful for you even if you are not doing object graphics. It can tell you if a point is exterior to a region, interior to a region, on edge or on vertex.

-Paul Sorenson

```
"Isa Usman" <eepisu@bath.ac.uk> wrote in message
news:bnu9ng$10h6@newton.cc.rl.ac.uk...

> Hello,

> I'm trying to get the indices to the points that fall within a rectangular

> region. I'm aware of David Fanning's INSIDE routine that does this but

> because I have got 100,000 points its going to involve a lot of looping.

So

> I decided to use the HIST_ND with reverse indices but its not giving me the

> right results. What am I missing?

> Thanks in advance.

> Isa
```

```
> IDL> x = [0.5,-0.5,-0.5,0.5]
> IDL> y = [0.5,0.5,-0.5,-0.5]
> IDL> z = transpose([[x],[y]])
> IDL> result = hist_nd(z,min = [0,0], max = [1,1], nbins = 1, reverse_indices
> = ri)
> print, ri[ri[0]:ri[1]-1]
> 0 1 2 3
>
```