Subject: Re: Histogram & Cumulative Distribution Functions
Posted by sdj on Mon, 30 Aug 2004 13:52:56 GMT
View Forum Message <> Reply to Message

Dear Justin,

Thanks for your help, your tip has indeed solved my problem.

FYI, I also found an alternative function for "value_locate" written
by Martin Schultz.

Regards,
Pepe

****
Pepe S. D. Juevara

- Risspekt de man and de nature - Ahi -
****

```
 ;------------------------------------------------------------ --
; Name: SEARCH (function)
;
; Purpose: Perform a binary search for the data point closest
;                  to a given value. Data must be sorted.
;
; Calling Sequence: index = SEARCH(data, value)
;
; Inputs:
;            data -> a sorted data vector
;            value -> the value to look for
;
; Outputs: The function returns the index of the nearest data point.
;
; Notes: This routine is much faster than WHERE or MIN for
;              large arrays. It was written in response to a
newsgroup
;              request by K.P. Bowman.
;
; Example:
;                test = findgen(10000)
;                print, search(test, 532.3)
;               ; prints 532
;
; Modification History: mgs, 21 Sep 1998: VERSION 1.00
;
;-
```

```
; Copyright (C) 1998, Martin Schultz, Harvard University
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine search"
 ;--------------------------------------------------------- --

FUNCTION search, data, value


    ; search first occurence of value in data set
    ; data must be sorted

    ; simple error checking on data and value
    if (n_elements(value) eq 0) then begin
       message,'Must supply sorted data array and value)',/CONT
       return, -1
    endif

    ndat = n_elements(data)

    try = fix(0.5*ndat)
    step = 0.5*try

    ; find index of nearest points
    while (step gt 1) do begin
       if (data[try] gt value) then $
          try = try-step $
       else   $
          try = try+step
       step = fix(0.5*(step+1))
    endwhile

    ; now get the data point closest to value
    ; can only be one out of three (try-1, try, try+1)
    dummy = min( abs(value-data[try-1:try+1]), location )

    return,try+location-1

end
```

Justin <kf1zr0y02@sneakemail.com> wrote in message
news:<Xns9552C28517E5kf1zr0y02sneakemail@18.181.0.25>...
> Ooops. Late on a Friday. I was meaning cdf in several places I wrote pdf.

> Still would have worked mind you. Soz.
>
> So if h is the output of HISTO then:
> cumul = TOTAL(h, /CUMULATIVE)
> tot = TOTAL(FLOAT(h))
> cdf = cumul/tot
>
> To find the 95th percentile use VALUE_LOCATE on the cdf to get the
> index of the array element closest to 0.95
>
> index = VALUE_LOCATE(cdf, 0.95)
>
> If 'l' contains the histo locations then your 95th percentile is at:
> l[index]
>
> Justin <kf1zr0y02@sneakemail.com> wrote in
> news:Xns9552C1E35BA22kf1zr0y02sneakemail@18.181.0.25:
>
>> To get the CDF from a (discrete) PDF use the TOTAL function with the
>> CUMULATIVE keyword:
>>
>> So if h is the output of HISTO then:
>> cumul = TOTAL(h, /CUMULATIVE)
>> tot = TOTAL(FLOAT(h))
>> pdf = cumul/tot
>>
>> To find the 95th percentile use VALUE_LOCATE on the pdf to get the
>> index of the array element closest to 0.95
>>
>> index = VALUE_LOCATE(pdf, 0.95)
>>
>> If 'l' contains the histo locations then your 95th percentile is at:
>> l[index]
>>
>> Make sure you have enough bins in the histogram otherwise the
>> percentile value can be coarse. You could even create a new histogram
>> (just for the cdf calculation) with nbins >= number of data points to
>> give an accuarate percentile value.
>>
>> Hope this helps,
>>
>> Justin