Subject: Re: hist_nd question
Posted by Wox on Mon, 05 Mar 2007 15:42:35 GMT
View Forum Message <> Reply to Message

Ok, it's not so simple. The problem is it can fall outside the
boundries for 1 dimension (-1 or nbins[i] ) and inside for others.
Since the result is summed to h, expanding the boundries with 1 will
not work. Any idea how to do this without using a where in the for
loop?

On Mon, 05 Mar 2007 16:17:09 +0100, Wox <nomail@hotmail.com> wrote:

> The fix is simple: change two lines.
>
> h=(nbins[s[0]-1]-1)<long((V[s[0]-1,*]-mn[s[0]-1])/bs[s[0]-1]) >0L
> =>
> h=nbins[s[0]-1]<long((V[s[0]-1,*]-mn[s[0]-1])/bs[s[0]-1])>-1L
>
> h=nbins[i]*h+((nbins[i]-1)<long((V[i,*]-mn[i])/bs[i])>0L)
> =>
> h=nbins[i]*h+(nbins[i]<long((V[i,*]-mn[i])/bs[i])>-1L)
>
>
>
> But since this function was written be JD and presented on Fanning's
> website, I obviously started doubting myself :-). Hence the question:
> is this a bug or a feature? And by changing these two lines, are there
> negative consequences which I overlooked?