Subject: Re: Generating N random numbers that add to a TOTAL
Posted by Russell Ryan on Thu, 07 Aug 2014 14:18:26 GMT
View Forum Message <> Reply to Message

You might have better luck with:

n_rand_var = 1000
requested_total=1000.

rand=randomu(seed,n_rand_var,/double)
rand*=(requested_total/total(rand))

This looks to be uniformly distributed, however it's not clear over what range it's uniform because the total(rand) in the denominator isn't necessarily the same. In the limit of n_rand_var -> infinity, then I think the total will converge to n_rand_var/2 and so the range will be

2*requested_total/n_rand_var

but for n_rand_var != infinity then it's a bit vague.

R

On Wednesday, August 6, 2014 11:52:47 PM UTC-4, Gianguido Cianci wrote:
> Hi all,
>
>
>
> I am wondering if anybody has suggestions on how to improve the function below. It seems ok for floating precision numbers.
>
>
>
> For integers it's a different story:
>
> It works great if N<<TOTAL. When N approaches TOTAL I get a few numbers and then a bunch of zeros... Also, setting /DIFFERENT makes it run for ever if N is large. Also, the sum of res adds up TOTAL=/-1, not always to TOTAL exactly...
>
>
>
> Suggestions?
>
>
>
> Thanks,

```
>
>  Gianguido
>
>
>
>
>
>
>
>  FUNCTION nrndaddto,  n,  total,  integers = integers,  different = different
>
>
>
>  compile_opt idl2
>
>
>
>  res = dblarr(n)
>
>  res[0] = randomu(seed, 1,  /double)*(total)
>
>
>
>  FOR i = 1,  n-2 DO BEGIN
>
>    res[i] = randomu(seed, 1,  /double)*(total-total(res[0:i-1],  /double))
>
>  ENDFOR
>
>  res[n-1] = total-total(res[0:n-2],  /double)
>
>
>
>  IF ~keyword_set(integers) THEN integers = 0
>
>
>
>  IF keyword_set(integers) THEN res = round(res)
>
>  IF keyword_set(different) THEN BEGIN
>
>    IF n_elements(res) NE n_elements(unique(res,  /sort)) THEN res = $
>
>    nrndaddto(n, total, integers = integers, different = 1)
>
>  ENDIF
>
>
```

```
>
>
>
> RETURN,  res
>
> END
```