

Re: [PHP] Deviation? Distribution? OT?

Source: <http://coding.derkeiler.com/Archive/PHP/php.general/2007-05/msg00054.html>

- *From:* parasane@xxxxxxxxxx ("Daniel Brown")
 - *Date:* Tue, 1 May 2007 17:48:17 -0400
-

I don't think I'm quite following why you wouldn't just want to break it up into groups of 20....

On 5/1/07, Richard Lynch <ceo@xxxxxxxxxx> wrote:

My boss and the web designer have decided to do something that requires statistical formulae well beyond my statistically-challenged capabilities, so I'm turning to y'all...

Basically, the current query looks something like this:

```
select * from
(select whatever, count(*) as popular
from the_table
group by whatever
order by popular desc
limit 100
) as pop
order by rand()
```

So basically it's the "Top 100" popular, ordered at random

They then want to have the MOST popular stuff with a CSS class="t5" and the least popular with class="t1"

The t5 are BIG AND BOLD and the t1 are >> tiny and plain << basically, with t2, t3, and t4 sort of "in between" in bigness and boldness.

The CSS guy has that bit worked out, in terms of style.

I just have to label the dang things t1 to t5, no more, no less.

Now I have NO IDEA what the range is going to be for the "popuclar" score as time goes on and the site gets super-popular...

So I guess I want some kind of standard deviation thingie among those

Re: [PHP] Deviation? Distribution? OT?

Top 100?

I know for sure I don't want just 20 of each t1 through t5.

There are way more t1 than there are t5.

So maybe it's more logarithmic than standard deviation?

Or is there some kind of weighted deviation?

Or maybe it's some other statistical thingie like standard deviation only not?

All I now for sure is, my current hack of taking `log($popular, 3)` in PHP is only going to sort of "work" with our sample data and make them THINK that it's working, when it's actually quite borked as soon as the number of data points increases. (I.e., next week or so)

Little help here?

I suspect it's going to turn out to be some magical MySQL statistics function, but not knowing what "standard deviation" actually *means* any more, I'm at a loss to guess which one...

I guess I *could* take the results in order without the wrapping `rand()` query, do min/max and log games in PHP on the highest/lowest, and then do a shuffle... But there's got to be a better way.

--

Some people have a "gift" link here.

Know what I want?

I want you to buy a CD from s