

Re: P-value from chi-square value: source code

Source: <http://sci.tech-archive.net/Archive/sci.stat.math/2005-04/msg00221.html>

- *From:* iandjmsmith@xxxxxxxx (Ian Smith)
 - *Date:* 12 Apr 2005 16:19:36 -0700
-

"Reef Fish" <Large_Nassau_Grouper@xxxxxxxx> wrote in message
news:<1113254955.951865.94110@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>...
> Ian Smith wrote:
>> "Reef Fish" <Large_Nassau_Grouper@xxxxxxxx> wrote in message
> news:<1113074906.743917.66500@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>...
>>> DZ wrote:
>>>> P.S. The p-value is $1 - \text{chi_cdf}(\text{ChiSquare}, \text{df})$
>>>>
>>>> If so, why do you need any ChiSquare software, let along coding it?
>>>> Why wouldn't any standard normal software suffice?
>>>>
>>>> DZ wrote:
>>>>> Dhruv <bhaveshrb@xxxxxxxx> wrote:
>>>>>> Ian Smith wrote:
>>>>>>> "Dhruv" <bhaveshrb@xxxxxxxx> wrote
>>>>>>> > Could anyone send me a link to the source code in C++ for
>>>>>>> > calculating the P-value from a chi-square value.
>>>>>>> > You are looking for code to evaluate the upper tail
> probability
>>>> of a
>>>>>>> chi-squared distribution.
>>>>>>> <http://odin.mdacc.tmc.edu/anonftp/#DCDFLIB>
>>>>>>>
>>>>>>> I actually tried the code and am having some problems with it.
> I
>>> am
>>>>>>> sort of a newbie to programming and though I have managed to
> write
>>>>>>> some good code, I am unable to figure out the problem at
> present.
>>>>>>> What I need basically is a function which takes in Chi-square
>>>> value
>>>>>>> and returns the P-value. Please help me on this.
>>>>>>>
>>>>>>> You need to compile the library and use the function "cdfchi" ,
>>>>>>> passing parameters as addresses, for example:
>>>>>>>
>>>>>>> double chi_cdf (double ChiSquare, double df) {
>>>>>>> double q, p, bound;

Re: P-value from chi-square value: source code

```
>>>>> int which=1, status;
>>>>> cdfchi(&which, &p, &q, &ChiSquare, &df, &status, &bound);
>>>>> return p;
>>>>> }
>>>
>>> And Ian Smith gave two links and said,
>>>
>>> IS> both have C code which will do it and there is nothing better
> out
>>> IS> there that I know of.
>>>
>>> Nothing better? Use either or both those links and let us know
>>> that the p-values (1-tail will do) for observed 1 df Chisquare
>>> values of 1, 10, and 100, to at least 8 significant figures,
>>> and then I'll tell you how many better ones there are. :-)
>>>
>>> -- Bob.
>>
> My point was merely my counterpoint to your "there is nothing better out
> there"
> even though you did have the qualifying disclaimer "that I know of".
>
```

Obviously I didn't qualify it enough for your liking. How about "I do not know of any C or C++ source code using double precision arithmetic which is better". From my point of view it is true and hopefully from yours it is no longer provocative!

```
>> For Gnumeric, the figures (rounded to 15 figures) you requested were
>> 3.17310507862914E-01, 1.56540225800255E-03 & 1.52397060483210E-23
>> I believe the 3rd one should be 1.52397060483211E-23
>
> For this pedantic numeric exercise, the p-value of 100 (1 d.f.) is
>
> 1.523970604832105213139466865031986167270080665559139211560717 E-23
>
> because I asked for only 60 significant figures of the 100 max. It
> took my 14-year old laptop less than 1 sec to get this (and the
> software
> ET (Electronic Tables) is one of the reason I kept the laptop. :-)
>
> 1.52397060483210E-23
> 1.52397060483211E-23
> You're correct that the 11 should be the rounded value.
>
> As for your software,
> http://members.aol.com/iandjmsmith/EXAMPLES.HTM (my own software, it
> gave
> a result incorrect in the last two significant figures:
>
> 1.5239706048321013 E-23
```

Re: P-value from chi-square value: source code

> 1.523970604832105213139466865031986167270080665559139211560717 E-23
>
> which suggests that those were your double-precision noise.
>

That's right, it's just the way Javascript displays numbers by default. Also it does only work with 64-bit arithmetic. You get no advantage from the fact that greater precision may be available with floating-point processors. It does mean you do not get an exaggerated sense of how good the accuracy is.

I don't think Javascript is meant to be a good language to calculate in. In fact, I would have thought it was a bad one because the code is slow and development in an environment with minimal support is tedious. It was chosen so that it can be run without needing any particular compiler but can easily be translated into the language of the user's choice should anyone decide the code is worth using.

The relative error is $2.6e-15$. Given the code essentially works with logs and exponentiates to get the answer, the minimum possible error bound (i.e. assuming that the call of `Math.exp` is the only source of error) which could be claimed is $6e-15$. By the way, not that I wish to compare algorithms, but really you should be looking to see how many figures have been lost not how many are remaining. An algorithm with 1 figure remaining from 7 is a lot better than an algorithm with 2 figures remaining from 16. I'm confident your 60 figures are correct because I would expect ET to get 97 or even 98 figures correct for that calculation with 100 figure arithmetic. There's no big deal in doing 100 figure arithmetic and the task is just the same ? minimise the accuracy being lost by using good algorithms.

>
>> However, I do not think an algorithm for a chi-squared distribution
>> for 1 degree of freedom (i.e. a simple transformation of the normal)
>> is of much interest.
>
> That was one point I made on DZ's post, before challenging you on the
> "nothing better out there" conjecture.
>
> For double-precision arithmetic, those software you named are probably
> as good any of dozens of others using double-precision with EXACT
> formulas for calculating the tail probabilities. Many of them are
> interpretive languages, such as S, S+, and so on, so that there is
> no NEED to do any CODING, especially using primitive codes like C,
> C+, C++ , etc.
>

I don't think interpretive languages such as S and S+ would be of much use to Dhruv, as I think he's looking to produce some C++ application, but it gives me the opportunity to mention that R (GNU S) has recently

Re: P-value from chi-square value: source code

changed its code for the cdf of the beta distribution to base it on TOMS708 (as per DCDFLIB) and its code for the cdf of the gamma distribution to use that of Gnumeric.

>
>> In my opinion, most statistical software available is pretty poor.
>
> Here you have to be careful about what do you mean by "most" and
> "pretty
> poor". There is no need to knock any software or tout your own -- all
> you have to do is to say what PRECISION it'll guarantee to deliver and
> let others decide whether it's good, good enough, or poor.
>
>

I should not have made that remark to help justify the choices I recommended and I take it back. To begin with I would guess most statistical software appears in analysis packages and I know little about them.

Part of the problem stems from sources such as A&S, NR and the like. Although they list properties of functions, they tend not to worry about the practical aspects of their use. The following is an example from NR ignoring the practical aspects of convergence of algorithms for the gamma distribution.

"Thus (6.2.5) and (6.2.7) together allow evaluation of the function for all positive a and x ." The paragraph is followed by code to evaluate the "incomplete gamma function $P(a, x)$ " and the "incomplete gamma function $Q(a, x) = 1 - P(a, x)$ ".

If you use fixed-precision arithmetic, the code which follows does not produce accurate answers for all positive a and x , no matter what fixed-precision you use! For any given a and x , fixed-precision arithmetic can deliver a suitably accurate answer but the required precision for the arithmetic will depend on the values of a and x .

I'm not going to apportion blame but there are programmers out there who seem to take them at their word and consequently create programs which don't behave as they expect. The old code in R2.0.1 is an example of just that, a problem soon to be of the past as R2.1.0 is due out soon.

Anyone looking for source code for the gamma distribution should watch out for this problem. Sometimes as with the R2.0.1 code, a normal approximation is used for large values. As long as this approximation is accurate enough and you don't want to use it for small shape parameters (as would be the case for Dhruv who just wants code for the Chi-squared distribution) then the code will be fine.

>> It would be nice when people recommend software if they included some

Re: P-value from chi-square value: source code

Re: P-value from chi-square value: source code

>> details of why. Admittedly, my explanation was perhaps too brief, but
>> at least I do know the software. I have looked at several software
>> packages in a variety of languages and I stick by what I said before.
>> There is nothing better out there that I know of.
>
> See my preceding comments. In my opinion, the software ET beats your
> software (in precision, flexibility, and versatility) as a modern jet
> beats a turtle in a 100 yard dash. :-)
>
> For Special functions, it has beta, incomplete beta, and incomplete
> gamma.
> It has all the trigonometric functions
> For continuous densities, it has the usual, in addition to
> Weibull, noncentral chi-square, double-noncentral t, double noncentral
> F,
> and others.
>
> Last but not least, it's NOT MY software, and I have absolutely
> nothing
> to do with it. :-) I was kindly given the copy by the author of the
> software when I was studying the accuracy of various exact and
> approximate
> numerical algorithms for probability functions.
>
> I don't know if its still commercially available.
>

ET may well have all the merits you describe but it does have some obvious drawbacks. Even if the source is C or C++ we have no reference. Without the source code I can't think of a convenient way in which Dhruv could call it from his C++ program.

There is a place for programs such as ET, mainly for generating test cases for assessing the performance of software such as DCDFLIB and Gnumeric. No matter how enthusiastic you are about ET, I find it hard to believe you would seriously recommend the inclusion of a high-precision implementation of algorithms for the chi-squared distribution.

> -- Bob
>>
>> If you wish to get a better idea of the accuracy of these methods try
>> the Chi-squared calculator at
>> <http://members.aol.com/iandjmsmith/EXAMPLES.HTM> (my own software is
>> not in C or I would have recommended it) or download Gnumeric and
> give
>> it a go!
>>
>> Ian Smith

Anyway a last word to Dhruv, I trust you have sorted your problems out

Re: P-value from chi-square value: source code

since we haven't heard any more about them!

Ian Smith

- **Follow-Ups:**

- ◆ **Re: P-value from chi-square value: source code**
◇ From: Dhruv

- **References:**

- ◆ **P-value from chi-square value: source code**
◇ From: Dhruv
- ◆ **Re: P-value from chi-square value: source code**
◇ From: Ian Smith
- ◆ **Re: P-value from chi-square value: source code**
◇ From: Dhruv
- ◆ **Re: P-value from chi-square value: source code**
◇ From: Reef Fish
- ◆ **Re: P-value from chi-square value: source code**
◇ From: Ian Smith
- ◆ **Re: P-value from chi-square value: source code**
◇ From: Reef Fish

- Prev by Date: **Re: Goodness of fit measures for a distribution**
- Next by Date: **Re: Goodness of fit measures for a distribution**
- Previous by thread: **Re: P-value from chi-square value: source code**
- Next by thread: **Re: P-value from chi-square value: source code**
- Index(es):
 - ◆ **Date**
 - ◆ **Thread**