# 140414 NTF Stirling’s Approximation

Note To File:

G. H. Boyle

14 April 2014

## References:

1. 140209 NTF How to write a function in Excel
2. 140409 NTF Discussion with Dr Yakovenko.
3. 140412 NTF Entropy in a Histogram.
4. 140414 Stirling’s Approximation.xlsm (a spreadsheet).
5. 140416 Entropy Equations with Stirling’s Approximation.

## Background

This note is written in an attempt to understand exactly how Stirling’s approximation to ln(A!) works.

## Stirling’s Approximation

This is “Stirling’s approximation” for ln(N!):

It is used in the formulae for entropy when N! is too large to be calculated.

## Standard Definition of Entropy

This is the standard equation for the definition of entropy:

where Ω is the total number of possible microstates associated with one configuration of a histogram:

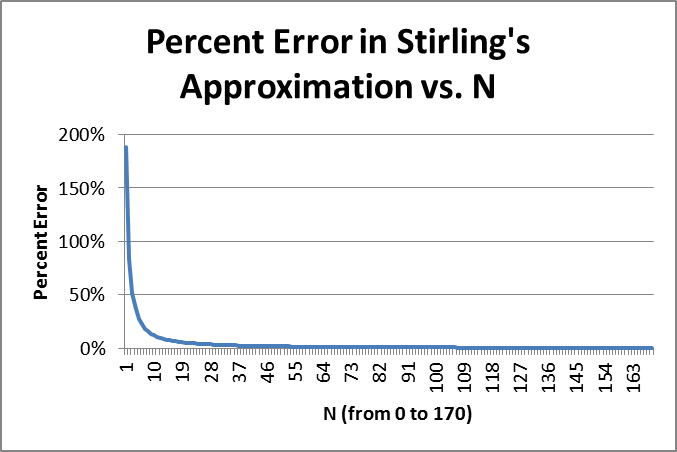
in a K-bin histogram.

But MS Excel cannot calculate ln(N!) if N > 170. That’s a pretty small number. To compare my analytical results (developed in MS Excel) with my experimental results (developed using C++ in the EiLab application) usefully, I need to use Stirling’s approximation whenever one of the following happens:

* N > 170; or
* Ni > 170.

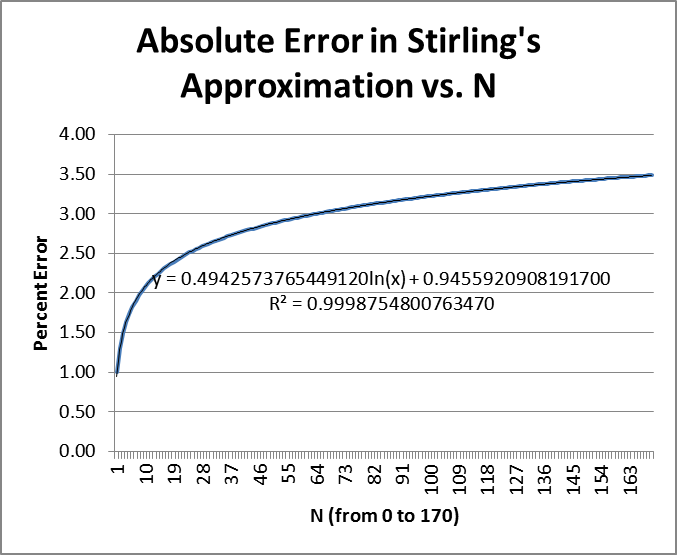
## Spreadsheet outputs

If you calculate ln(N!) for N in [0, 170], which can be done exactly, and Stirling’s approximation for the same N, and take the difference as a % of ln(N!), you get an interesting graph:



When you get to N = 170 the relative error is 0.005%. The absolute error is rising, but the value of ln(N!) is rising faster, so the relative error continues to decline as N gets larger. But for small N, in the area where my ABMs will be active, the relative error in Stirling’s approximation can be quite large.

BUT, a graph of the absolute error over the same domain is also interesting:

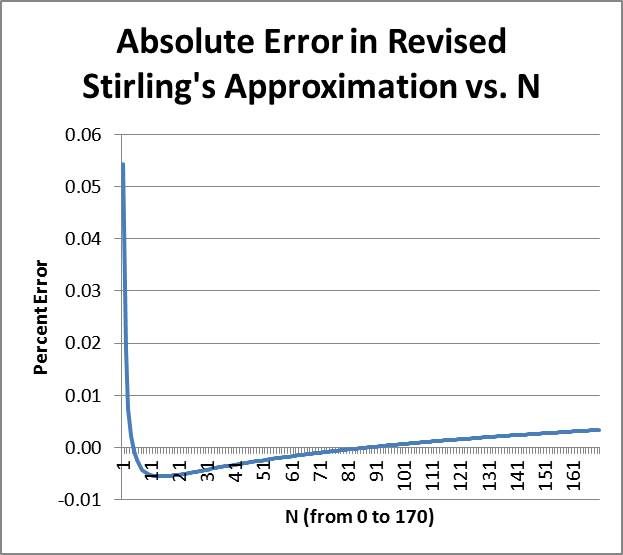
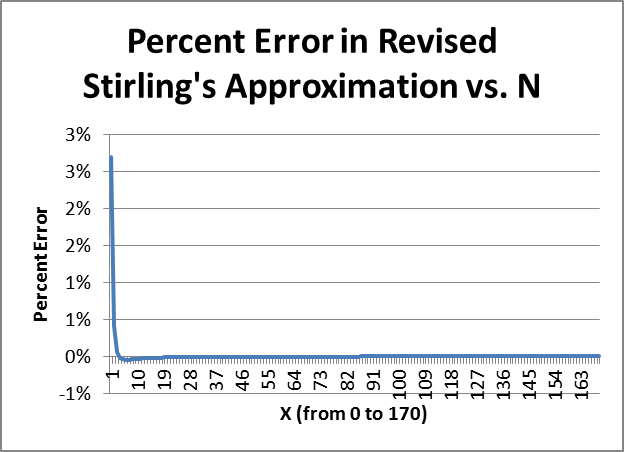


The blue line is the absolute error. The black line is a logarithmic trend line produced by MS Excel, with the equation for the trend line. I altered the formatting of the equation of the trend line to give me 16 digits of accuracy. Note that the R-squared value is very high.

This gives me an idea for a variation on Stirling’s approximation that might be more useful for small values of N.

I note that these numbers are absurdly precise and could possibly be replaced with 0.5 and 1.0 respectively, like this:

But, they are what they are.



A couple of tables will clarify. Looking at the results when N = 169 and 170, for both Stirling and revised Stirling, we get these two tables:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **LN(FACT(N))** | **Stirling’s Approximation** | **Absolute Error** | **Relative Error** |
| 169 | 701.4372638 | 697.9528828 | 3.484380987 | 0.004967488 |
| 170 | 706.5730622 | 703.0857343 | 3.487327947 | 0.004935552 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **LN(FACT(N))** | **Revised Stirling’s Approximation** | **Absolute Error** | **Relative Error** |
| 169 | 701.4372638 | 701.4339652 | 0.003298615 | 4.70265E-06 |
| 170 | 706.5730622 | 706.5697327 | 0.003329594 | 4.71231E-06 |

With the Stirling approximation of ln(N!) the absolute error is rising but the relative error is falling. For the revised version of Sterling’s approximation, both the absolute and relative error are rising. This is not good.

But, curiously, if I use the two simple parameters, things are a little better.

Using 0.5 and 1.0 as the two revised parameters:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **N** | **LN(FACT(N))** | **Revised Stirling’s Approximation** | **Absolute Error** | **Relative Error** |
| 169 | 701.4372638 | 701.5178 | -0.080568371 | -0.000114862 |
| 170 | 706.5730622 | 706.6536 | -0.080571271 | -0.000114031 |

The absolute error is larger and climbing, not good, but the relative error is larger but falling, good. In either case, when N ≈ 170, both sorts of revised formulae for Sterling’s approximation seem to perform better.

Here’s another way to summarize it:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Option Number** | **Option** | **Absolute Error at N=170** | **Relative Error at N=170** | **RMS Error on [0,170]** |
| 1 | Stirling unrevised | rising | falling | 3.044133 |
| 2 | Revised Stirling with precise parms | rising | rising | 0.003146 |
| 3 | Revised Stirling with {0.5, 1} as parms | rising | falling | 0.078870 |

The RMS error is computed as

The unrevised Sterling formula performs badly on the interval [0, 170] compared to the revised versions.

In summary:

* In my EiLab studies, I should avoid Stirling’s approximation, as I have a lot of bins with zero or 1 agent, and this is where Stirling’s approximation, in all of its forms, performs worst.
* It appears that Stirling’s approximation is not good for very small x, but I can use an exact formula in that part of the domain. I do not need to approximate there.
* A revised version of Stirling’s approximation seems to perform better for numbers of the order of magnitude of 170, where many ABMs will be functioning, and for which MS Excel cannot provide exact support in analysis via an approximation formula.
* For very large numbers (not shown here) Stirling’s approximation is good because the relative error declines as ln(N)/N and goes to zero as N rises. The issue is, then, is Stirling’s approximation, or a revised version, close enough for the domain of numbers in which most ABMs will operate? Certainly, in my ABMs, the number of agents is usually less than 1000, and so the following approach seems good.
* Use an exact calculation of ln(N!) when N < 170; and
* Use a revised Stirling’s approximation for N >= 170.

Then, I can implement this in both MS Excel when I do my analysis and also in C++ when I build my models.