March 25, 2021

MAGIC CONSTANT IN *HOW TO MEASURE ANYTHING IN CYBERSECURITY RISK*

ERIC BOWMAN

This is an investigation into the properties of lognormal distributions, to illuminate a mystery found in Chapter 3 of *How To Measure Anything in Cybersecurity.*

There is a mysterious constant in the Chapter 3 Excel spreadsheet, in column `J`, "Expected Inherent Loss": 3.28971, that is never explained. Some searching uncovers some clues but it's leaves questions.

In this paper, we will work through the math to uncover what this constant means, and why it is hard-coded.

The normal distribution has a probability density function defined as:

$$f_Z(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{1}$$

A random variable X with a log-normal distribution has the property that the $logX$ has a normal distribution. Using a log-normal distribution tends to reflect the domain, since a log-normally distributed variable X will never be negative, and tends to have a low-probability *long tail*. This makes sense, for example, to model the number of break-in attempts of the financial loss due to a security issue. The values are 0 or great, there tends to be a relatively modest expectation value, but there is a finite probability the numbers will be large.

The log-normal probability density function is defined as:

$$f_X(x) = \frac{1}{x} \cdot \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{1}{2}(\frac{\ln x-\mu}{\sigma})^2} \tag{2}$$

The ProbOnto website characterizes R's implementation of the normal and lognormal distributions as "lognormal1." These are represented in R using the `dnorm` and `dlnorm` funtions, respectively.

The book asks the estimator to choose a lower bound L and an upper bound U such that the estimator believes there is a 90% chance the observed value will be within that range. This means that L describes the 5th percentile of distribution, and U describes the 95th percentile.

For a log-normal distribution, the integral of the probability density function, called the cumulative distribution function (CDF) is described in terms of the CDF for a normal distribution: (in R, `plnorm`) as:

$$F_X(x) = \Phi(\frac{\log x - \mu}{\sigma}) \tag{3}$$

In R, this is represented as `plnorm`.

The challenge is to demonstrate that, given a particular value for L and U, the parameters μ and σ are defined.

If this is true, then we can solve for μ and σ in the following system of equations:

$$F_X(L, \mu, \sigma) = 0.05 \tag{4}$$

$$F_X(U, \mu, \sigma) = 0.95 \tag{5}$$

... which is the same as

$$\Phi(\frac{\log L - \mu}{\sigma}) = 0.05 \tag{6}$$

$$\Phi(\frac{\log U - \mu}{\sigma}) = 0.95 \tag{7}$$

which leads to

$$\frac{\log L - \mu}{\sigma} = \Phi^{-1}(0.05) \tag{8}$$

$$\frac{\log U - \mu}{\sigma} = \Phi^{-1}(0.95) \tag{9}$$

$$\log L - \mu = \sigma \cdot \Phi^{-1}(0.05) \tag{10}$$

$$\log U - \mu = \sigma \cdot \Phi^{-1}(0.95) \tag{11}$$

...where Φ^{-1} is the inverse CDF, also called the *quantile function* (in R, `qnorm`).

Adding these together lets us solve for μ:

$$
\begin{aligned}
(\log L + \log U) - 2\mu &= 2\sigma(\Phi^{-1}(0.05) + \Phi^{-1}(0.95)) \\
-2\mu &= 2\sigma(\Phi^{-1}(0.05) + \Phi^{-1}(0.95)) - (\log L + \log U) \\
\mu &= \frac{\log L + \log U - \sigma(\Phi^{-1}(0.05) + \Phi^{-1}(0.95))}{2} \\
\mu &= \frac{\log L + \log U}{2}
\end{aligned} \tag{12}
$$

In the last step, we are able to remove the Φ term because the normal distribution is symmetric, so we know that $\Phi(0.05) = -\Phi(0.95)$. We can approximately confirm this in R:

¡¡¿¿= qnorm(0.05) + qnorm(0.95) @

Subtracting them lets us solve for σ:

$$
\begin{aligned}
\frac{\log L - \mu}{\sigma} - \frac{\log U - \mu}{\sigma} &= \Phi^{-1}(0.05) - \Phi^{-1}(0.95) \\
\log L - \mu - (\log U - \mu) &= \sigma(\Phi^{-1}(0.05) - \Phi^{-1}(0.95)) \\
\sigma &= \frac{\log L - \log U}{\Phi^{-1}(0.05) - \Phi^{-1}(0.95)} \\
\sigma &= \frac{\log U - \log L}{2\Phi^{-1}(0.95)}
\end{aligned} \tag{13}
$$

...which is the result.

There is no closed form expression for Φ^{-1}. Therefore, the author uses the hard-coded value, though we could use an excel function to compute it, which would increase clarity. In Excel, this is the `NORM.INV` function, documented here.

In R,

¡¡¿¿= 2*qnorm(0.95) @ The corresponding evaluation in Excel, `=2*NORM.INV(0.95,0,1)` yields 3.289707254.

Email address: `eric.bowman@tomtom.com`