

*University of Texas, MD Anderson Cancer  
Center*

UT MD Anderson Cancer Center Department of Biostatistics  
Working Paper Series

---

*Year 2011*

*Paper 71*

---

Random inequalities between survival and  
uniform distributions

John D. Cook\*

\*M. D. Anderson Cancer Center, [cook@mdanderson.org](mailto:cook@mdanderson.org)

This working paper is hosted by The Berkeley Electronic Press (bepress) and may not be commercially reproduced without the permission of the copyright holder.

<http://biostats.bepress.com/mdandersonbiostat/paper71>

Copyright ©2011 by the author.

# Random inequalities between survival and uniform distributions

John D. Cook

## **Abstract**

This note will look at ways of computing  $P(X > Y)$  where  $X$  is a distribution modeling survival (gamma, inverse gamma, Weibull, log-normal) and  $Y$  has a uniform distribution. Each of these can be computed in closed form in terms of common statistical functions. We begin with analytical calculations and then include software implementations in R to make some of the details more explicit. Finally, we give a suggestion for using simulation to compute random inequalities that cannot be computed in closed form.

# Random inequalities between survival and uniform distributions

John Cook

September 14, 2011

## Abstract

This note will look at ways of computing

$$P(X > Y)$$

where  $X$  is a distribution modeling survival (gamma, inverse gamma, Weibull, log-normal) and  $Y$  has a uniform distribution. Each of these can be computed in closed form in terms of common statistical functions. We begin with analytical calculations and then include software implementations in R to make some of the details more explicit. Finally, we give a suggestion for using simulation to compute random inequalities that cannot be computed in closed form.

## 1 Analytical results

For any distributions on independent random variables  $X$  and  $Y$ ,

$$P(X > Y) = \int_{-\infty}^{\infty} f_X(x)F_Y(x) dx.$$

Here we use  $f_W$  and  $F_W$  for the PDF and CDF functions of  $W$  respectively. For more information on such inequalities, see [1].

Assume  $Y$  is uniformly distributed on  $[a, b]$  with  $0 \leq a < b < \infty$ . Then

$$F_Y(x) = \begin{cases} 0 & \text{if } x < a \\ (x - a)/(b - a) & \text{if } a \leq x \leq b \\ 1 & \text{if } x > b \end{cases}$$

and it follows that

$$\begin{aligned} \int_{-\infty}^{\infty} f_X(x)F_Y(x) dx &= \int_a^b \frac{x - a}{b - a} f_X(x) dx + \int_b^{\infty} f_X(x) dx \\ &= \frac{1}{b - a} \int_a^b x f_X(x) dx - \frac{a}{b - a} (F_X(b) - F_X(a)) + 1 - F_X(b) \\ &= \frac{1}{b - a} \int_a^b x f_X(x) dx + 1 - \frac{1}{b - a} (b F_X(b) - a F_X(a)) \end{aligned}$$

Assuming software for evaluating  $F_X$  is available, the problem of computing  $P(X > Y)$  reduces to the problem of computing

$$\int_a^b x f_X(x) dx$$

## 2 $X \sim$ gamma

Assume  $X$  has a gamma distribution with shape parameter  $\alpha$ . Without loss of generality, we may assume  $X$  has scale parameter 1. Otherwise if  $X$  had scale  $\beta$ , one could compute

$$P(X > Y) = P(X/\beta > Y/\beta)$$

where  $Y/\beta$  is uniform on  $[a/\beta, y/\beta]$ .

Now

$$f_X(x) = \frac{1}{\Gamma(\alpha)} x^{\alpha-1} \exp(-x)$$

and

$$\begin{aligned} \int_a^b x f_X(x) dx &= \frac{\Gamma(\alpha+1)}{\Gamma(\alpha)} \frac{1}{\Gamma(\alpha+1)} \int_a^b x^{\alpha+1-1} \exp(-x) dx \\ &= \alpha (F_W(b) - F_W(a)) \end{aligned}$$

where  $W$  is a gamma random variable with shape  $\alpha + 1$ .

## 3 $X \sim$ inverse gamma

Now assume  $X$  has an inverse gamma distribution with shape  $\alpha$ . As before, we can assume without loss of generality that the shape parameter is 1.

Now

$$f_X(x) = \frac{1}{\Gamma(\alpha)} x^{-\alpha-1} \exp(-1/x) dx$$

and

$$\begin{aligned} \int_a^b x f_X(x) dx &= \frac{1}{\Gamma(\alpha)} \int_a^b x^{-\alpha} \exp(-1/x) dx \\ &= \frac{1}{\Gamma(\alpha)} \int_{1/b}^{1/a} u^{\alpha-2} \exp(-u) du \\ &= \frac{\Gamma(\alpha-1)}{\Gamma(\alpha)} \int_{1/b}^{1/a} \frac{1}{\Gamma(\alpha-1)} u^{\alpha-1-1} \exp(-u) du \\ &= \frac{1}{\alpha-1} (F_W(1/a) - F_W(1/b)) \end{aligned}$$

where  $W$  is a gamma random variable with shape  $\alpha - 1$ .

## 4 $X \sim$ Weibull

Now assume  $X$  has a Weibull distribution. As with the case of the gamma and inverse distributions, we can assume without loss of generality that  $X$  has scale 1.

Now

$$f_X(x) = \beta x^{\beta-1} \exp(-x^\beta)$$

and

$$\begin{aligned} \int_a^b x f_X(x) dx &= \beta \int_a^b x^\beta \exp(-x^\beta) dx \\ &= \int_{a^\beta}^{b^\beta} u \exp(-u) u^{\frac{1}{\beta}-1} du \\ &= \Gamma(1 + 1/\beta) \int_{a^\beta}^{b^\beta} \frac{1}{\Gamma(1 + 1/\beta)} u^{1/\beta} \exp(-u) du \\ &= \Gamma(1 + 1/\beta) (F_W(b^\beta) - F_W(a^\beta)) \end{aligned}$$

where  $W$  is a gamma random variable with shape  $1 + 1/\beta$ .

## 5 $X \sim$ log-normal

Assume  $X$  has a log-normal distribution. Then

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \frac{1}{x} \exp(-(\log x - \mu)^2/2\sigma^2)$$

and

$$\begin{aligned} \int_a^b x f_X(x) dx &= \frac{1}{\sqrt{2\pi}\sigma} \int_a^b \exp(-(\log x - \mu)^2/2\sigma^2) dx \\ &= \frac{1}{\sqrt{2\pi}\sigma} \int_{\log a}^{\log b} \exp(-(u - \mu)^2/2\sigma^2) \exp(u) du \end{aligned}$$

Now

$$u - \frac{(u - \mu)^2}{2\sigma^2} = -\frac{(u - (\mu + \sigma^2))^2}{2\sigma^2} + \mu + \frac{\sigma^2}{2}$$

and so the integral above becomes

$$\exp(\mu + \sigma^2/2) \frac{1}{\sqrt{2\pi}\sigma} \int_{\log a}^{\log b} \exp\left(-\frac{(u - (\mu + \sigma^2))^2}{2\sigma^2}\right) du$$

which equals

$$\exp(\mu + \sigma^2/2) \left( \Phi\left(\frac{\log b - \mu - \sigma^2}{\sigma}\right) - \Phi\left(\frac{\log a - \mu - \sigma^2}{\sigma}\right) \right).$$

## 6 R implementation

The following R functions compute the random inequalities described in the previous section. Note that they properly avoid taking the logarithm or reciprocal of zero.

```
gamma.ineq <- function(shape, scale, a, b)
{
  a <- a/scale
  b <- b/scale

  ineq <- 1 - (b*pgamma(b, shape) - a*pgamma(a, shape)) / (b - a)
  ineq <- ineq + shape*(pgamma(b, shape+1) - pgamma(a, shape+1))/(b - a)
  return( ineq )
}

inverse.gamma.ineq <- function(shape, scale, a, b)
{
  a <- a/scale
  b <- b/scale

  # If X ~ IG(shape) then 1/X ~ gamma(shape)
  # P(X < c) = P(1/X > 1/c)
  cdf <- function(x) { return(pgamma(1/x, shape, lower.tail = FALSE)) }

  ineq <- 1 - (b*cdf(b) - a*cdf(a))/(b - a)
  upper <- ifelse(a > 0, pgamma(1/a, shape-1), 1)
  ineq <- ineq + (upper - pgamma(1/b, shape-1)) / ((shape-1)*(b-a))
  return( ineq )
}

weibull.ineq <- function(shape, scale, a, b)
{
  a <- a/scale
  b <- b/scale

  ineq <- 1 - (b*pweibull(b, shape) - a*pweibull(a, shape))/(b - a)
  arg <- 1 + 1/shape
  ineq <- ineq + gamma(arg)*(pgamma(b^shape, arg) - pgamma(a^shape, arg))/(b - a)
  return( ineq )
}

log.normal.ineq <- function(mu, sigma, a, b)
{
  ineq <- 1 - (b*plnorm(b, mu, sigma) - a*plnorm(a, mu, sigma))/(b - a)
}
```

```

phi <- function(x) { return( pnorm((log(x) - mu - sigma^2)/sigma) ) }

lower <- ifelse(a > 0, phi(a), 0)
ineq <- ineq + exp(mu + sigma^2/2)*(phi(b) - lower)/(b-a)
return( ineq )
}

```

These functions were tested by comparing their results to the corresponding simulation results.

## 7 Simulation

Let  $X$  and  $Y$  be any independent random variables. Let  $B$  be the Bernoulli( $p$ ) random variable that is 1 if  $X > Y$  and zero otherwise. We wish to estimate  $p$  by repeatedly sampling from the distributions of  $X$  and  $Y$ . If we knew  $p$ , we could determine the number of samples necessary to meet a specified accuracy. The standard error in estimating  $p$  from  $n$  samples is  $\sqrt{p(1-p)/n}$  and so if we an approximately 0.95 probability of estimating  $p$  with relative error less than  $\varepsilon$ , we need to solve

$$\frac{\sqrt{p(1-p)/n}}{p} = \frac{\varepsilon}{2}$$

for  $n$ . This says

$$n = \frac{4(1-p)}{p\varepsilon^2}.$$

One could first estimate  $p$  from a small sample and use  $\hat{p}$  from that sample to estimate the number of further samples needed to meet the accuracy goal. The following R code will carry out this procedure given functions that generate random samples from  $X$  and  $Y$ .

```

sample.ineq <- function(xrng, yrng, epsilon)
{
  sample.size1 <- 1000
  max.sample.size <- 1000000

  x <- xrng(sample.size1)
  y <- yrng(sample.size1)
  count1 <- sum( x > y )
  p.hat1 <- count1 / sample.size1

  n <- 4*(1 - p.hat1)/(p.hat1*epsilon^2)
  if (n < sample.size1) return(p.hat1)

  sample.size2 = min(n, max.sample.size)
}

```

```

x <- xrng(sample.size2)
y <- yrng(sample.size2)
count2 <- sum( x > y )

# count2 / sample.size2 should be accurate enough (with prob 0.95)
# but we might as well take advantage of the first sample.
return( (count1 + count2)/(sample.size1 + sample.size2) )
}

```

The following examples use `sample.ineq` to test the functions in the previous section.

```

test.helper <- function(a, b, integration, xrng)
{
  yrng <- function(n) {return( runif(n, a, b) )}
  simulation <- sample.ineq(xrng, yrng, 0.001)
  cat("By integration: ", integration, "\n")
  cat("By simulation: ", simulation, "\n\n")
}

cat("Testing P(X > Y) where X ~ gamma(3, 10) and Y ~ unif(15, 45)\n")
integration <- gamma.ineq(3, 10, 15, 45)
xrng <- function(n) {return( rgamma(n, 3, scale=10) )}
test.helper(15, 45, integration, xrng)

cat("Testing P(X > Y) where X ~ inverse gamma(3, 1) and Y ~ unif(0.5, 0.8)\n")
integration <- inverse.gamma.ineq(3, 1, 0.5, 0.8)
xrng <- function(n) { return( 1/rgamma(n, shape = 3) ) }
test.helper(0.5, 0.8, integration, xrng)

cat("Testing P(X > Y) where X ~ Weibull(4, 1) and Y ~ unif(0.8, 1.2)\n")
integration <- weibull.ineq(4, 1, 0.8, 1.2)
xrng <- function(n) { return( rweibull(n, 4) ) }
test.helper(0.8, 1.2, integration, xrng)

cat("Testing P(X > Y) where X ~ log normal(3, 2) and Y ~ unif(0, 8)\n")
integration <- log.normal.ineq(3, 2, 0, 8)
xrng <- function(n) { return( rlnorm(n, 3, 2) ) }
test.helper(0, 8, integration, xrng)

```

These tests produced the following output.

```

Testing P(X > Y) where X ~ gamma(3, 10) and Y ~ unif(15, 45)
By integration: 0.4480053
By simulation: 0.4480663

```



Testing  $P(X > Y)$  where  $X \sim \text{inverse gamma}(3, 1)$  and  $Y \sim \text{unif}(0.5, 0.8)$   
By integration: 0.2095968  
By simulation: 0.2097247

Testing  $P(X > Y)$  where  $X \sim \text{Weibull}(4, 1)$  and  $Y \sim \text{unif}(0.8, 1.2)$   
By integration: 0.3771884  
By simulation: 0.3774764

Testing  $P(X > Y)$  where  $X \sim \text{log normal}(3, 2)$  and  $Y \sim \text{unif}(0, 8)$   
By integration: 0.8061174  
By simulation: 0.8068019

## 8 Reference

- [1] “Numerical Computation of Stochastic Inequality Probabilities” (August 2008). UT MD Anderson Cancer Center Department of Biostatistics Working Paper Series. Working Paper 46. <http://www.bepress.com/mdandersonbiostat/paper46>.

