Journal of Statistical Software

August 2006, Volume 16, Code Snippet 2. *http://www.jstatsoft.org/*

R Programs for Computing Truncated Distributions

Saralees Nadarajah
University of Manchester

Samuel Kotz
George Washington University

Abstract

Truncated distributions arise naturally in many practical situations. In this note, we provide programs for computing six quantities of interest (probability density function, mean, variance, cumulative distribution function, quantile function and random numbers) for *any* truncated distribution: whether it is left truncated, right truncated or doubly truncated. The programs are written in R: a freely downloadable statistical software.

Keywords: truncated distributions, R.

1. Introduction

Truncated distributions arise in many practical situations, particularly in numerous industrial settings (Cho and Govindaluri 2002; Jeang 1997; Kapur and Cho 1994, 1996; Phillips and Cho 1998, 2000; Khasawneh, Bowling, Kaewkuekool, and Cho 2004, 2005). Final products are often subject to screening inspection before being sent to the customer. The usual practice is that if a product's performance falls within certain tolerance limits, it is judged conforming and sent to thee customer. If it fails, a product is rejected and thus scrapped or reworked. In this case, the actual distribution to the customer is truncated. Another example can be found in a multistage production process, in which inspection is performed at each production stage. If only conforming items are passed on to the next stage, the actual distribution is a truncated distribution. Accelerated life testing with samples censored is also a good example. In fact, the concept of a truncated distribution plays a significant role in analyzing a variety of production processes, process optimization and quality improvement.

Truncated distributions can also be used to model intensity statistics in the study of atomic heterogeneity (Bhowmick, Mukhopadhyay, and Mitra 2000). The justification being that: 1) atomic heterogeneity led to the intensity statistics being modified from Gaussian to near Gaussian forms (Shmueli 1979; Shmueli and Wilson 1981); and 2) in reality, the structure factors or normalized structure factors do not range from $-\infty$ to ∞ but over a finite range.

Another situation arises with respect to high-performance Ethernet, where measurements can match well a truncated distribution, with a much better fit over smaller file/request sizes then the commonly used Pareto distribution. Field, Harder, and Harrison (2004) showed that measured traffic from three locations on a state-of-the-art switched Ethernet fit closely various truncated distributions.

The aim of this note is to study the truncated version of any given distribution: left truncated, right truncated or the doubly truncated version. We provide programs in R for computing six quantities of interest for the truncated distribution. The programs are written in R (R Development Core Team 2006, `http://www.R-project.org/`) because, unlike other statistical software, it is freely downloadable from the Internet at `http://CRAN.R-project.org/`, see also Ihaka and Gentleman (1996). The programs are written in such a way to accept any value for the truncation points or any distribution.

2. Programs

Suppose we have a continuous distribution with probability density function (pdf) and cumulative distribution function (cdf) specified by $g(\cdot)$ and $G(\cdot)$, respectively. Let X be a random variable representing the truncated version of this distribution over the interval $[a, b]$, where $-\infty < a < b < \infty$. The pdf, mean, variance, cdf, quantile function and the n random numbers of X are given by

$$f_X(x) = \begin{cases} \dfrac{g(x)}{G(b) - G(a)}, & \text{if } a \le x \le b, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

$$\mathrm{E}(X) = \int_a^b x f_X(x) dx, \tag{2}$$

$$\mathrm{Var}(X) = \int_a^b \{x - \mathrm{E}(X)\}^2 f_X(x) dx, \tag{3}$$

$$F_X(x) = \frac{G(\max(\min(x, b), a)) - G(a)}{G(b) - G(a)}, \tag{4}$$

$$F_X^{-1}(p) = G^{-1}\left(G(a) + p\left(G(b) - G(a)\right)\right) \tag{5}$$

and

$$x_i = F_X^{-1}\left(u_i\right), \tag{6}$$

respectively, where u_i, $i = 1, 2, \ldots, n$ are n uniform $(0, 1)$ random numbers.

The functions in R for computing (1)–(6) are given below and in the file ‘`truncated.R`’, published with this paper. The calling sequence of the functions and their return values are noted in Table 1. The character string `spec` specifies the forms for $g(\cdot)$ and $G(\cdot)$. For instance, if `spec = "norm"` then $g(\cdot)$ and $G(\cdot)$ will correspond to the standard normal distribution. If `spec = "beta"` then $g(\cdot)$ and $G(\cdot)$ will correspond to the beta distribution and its two shape parameters will have to be supplied as additional arguments.

The default values of the arguments `a` and `b` are set to `-Inf` and `Inf`, respectively. This means that the functions correspond to the untruncated case by default.

The functions `extrunc()` and `vartrunc()` use the `integrate()` function to perform the integration in equations (2) and (3). The standard arguments used for the `integrate()` function are passed as arguments for `extrunc()` and `vartrunc()`. This way, the user has complete control over the accuracy and the stability of the results. If these arguments are not passed as arguments then the default values will be used, see the R documentation on `integrate()` for the default values.

Calling sequence	Value
`dtrunc(x, spec, a, b, ...)`	$f_X(x)$ in (1)
`extrunc(spec, a, b, ...)`	$\mathrm{E}(X)$ in (2)
`vartrunc(spec, a, b, ...)`	$\mathrm{Var}(X)$ in (3)
`ptrunc(x, spec, a, b, ...)`	$F_X(x)$ in (4)
`qtrunc(p, spec, a, b, ...)`	$F_X^{-1}(p)$ in (5)
`rtrunc(n, spec, a, b, ...)`	$x_i = F_X^{-1}(u_i)$ in (6)

Table 1: Calling sequence and value for the truncated distribution.

The function `dtrunc()` presented below implements (1) for given `x` and a distribution specification `spec` on the interval (`a`, `b`) defaulting to $(-\infty, \infty)$.

```
dtrunc <- function(x, spec, a = -Inf, b = Inf, ...)
{
    tt <- rep(0, length(x))
    g <- get(paste("d", spec, sep = ""), mode = "function")
    G <- get(paste("p", spec, sep = ""), mode = "function")
    tt[x>=a & x<=b] <- g(x[x>=a&x<=b], ...)/(G(b, ...) - G(a, ...))
    return(tt)
}
```

The function `extrunc()` presented below implements (2) for given distribution specification `spec` on the interval (`a`, `b`) defaulting to $(-\infty, \infty)$.

```
extrunc <- function(spec, a = -Inf, b = Inf,...)
{
    f <- function(x) x * dtrunc(x, spec, a = a, b = b, ...)
    return(integrate(f, lower = a, upper = b)$value)
}
```

The function `vartrunc()` presented below implements (3) for given distribution specification `spec` on the interval (`a`, `b`) defaulting to $(-\infty, \infty)$.

```
vartrunc <- function(spec, a = -Inf, b = Inf, ...)
```

```
{
    ex <- extrunc(spec, a = a, b = b, ...)
    f <- function(x) (x - ex)^2 * dtrunc(x, spec, a = a, b = b, ...)
    tt <- integrate(f, lower = a, upper = b)$value
    return(tt)
}
```

The function `ptrunc()` presented below implements (4) for given `x` and a distribution specification `spec` on the interval (`a`, `b`) defaulting to $(-\infty, \infty)$.

```
ptrunc <- function(x, spec, a = -Inf, b = Inf, ...)
{
    tt <- x
    aa <- rep(a, length(x))
    bb <- rep(b, length(x))
    G <- get(paste("p", spec, sep = ""), mode = "function")
    tt <- G(apply(cbind(apply(cbind(x, bb), 1, min), aa), 1, max), ...)
    tt <- tt - G(aa, ...)
    tt <- tt/(G(bb, ...) - G(aa, ...))
    return(tt)
}
```

The function `qtrunc()` presented below implements (5) for given `p` and a distribution specification `spec` on the interval (`a`, `b`) defaulting to $(-\infty, \infty)$.

```
qtrunc <- function(p, spec, a = -Inf, b = Inf, ...)
{
    tt <- p
    G <- get(paste("p", spec, sep = ""), mode = "function")
    Gin <- get(paste("q", spec, sep = ""), mode = "function")
    tt <- Gin(G(a, ...) + p*(G(b, ...) - G(a, ...)), ...)
    return(tt)
}
```

The function `rtrunc()` presented below implements (6) for given `n` and a distribution specification `spec` on the interval (`a`, `b`) defaulting to $(-\infty, \infty)$.

```
rtrunc <- function(n, spec, a = -Inf, b = Inf, ...)
{
    x <- u <- runif(n, min = 0, max = 1)
    x <- qtrunc(u, spec, a = a, b = b,...)
    return(x)
}
```

Note that the functions `dtrunc()`, `ptrunc()` and `qtrunc()` accept vector values for their first arguments.

3. Example 1

This example computes truncated versions of the standard normal density function for $-a = b = 0.5$, $-a = b = 1$, $-a = b = 2$, and $-a = b = 2.5$. The function `dtrunc()` from Table 1 is used. The argument `spec` is taken to be `"norm"`. The plot of the computed densities is shown in Figure 1.

```
R> x <- seq(-3, 3, by = 0.1)
R> y1 <- dnorm(x)
R> y2 <- dtrunc(x, "norm", a = -0.5, b = 0.5, mean = 0, sd = 2)
R> y3 <- dtrunc(x, "norm", a = -1, b = 1, mean = 0, sd = 2)
R> y4 <- dtrunc(x, "norm", a = -2, b = 2, mean = 0, sd = 2)
R> yrange <- range(y1, y2, y3, y4)
R> plot(x, y1, type = "l", xlab = "x", ylab = "PDF", xlim = c(-3,
+     3), ylim = yrange)
R> lines(x, y2, lty = 2)
R> lines(x, y3, lty = 3)
R> lines(x, y4, lty = 4)
```

In Figure 1, the solid curve corresponds to the standard normal pdf, the curve of lines corresponds to the truncated version with $-a = b = 0.5$, the curve of dots corresponds to the truncated version with $-a = b = 1$, and the curve of lines and dots corresponds to the truncated version with $-a = b = 2$.

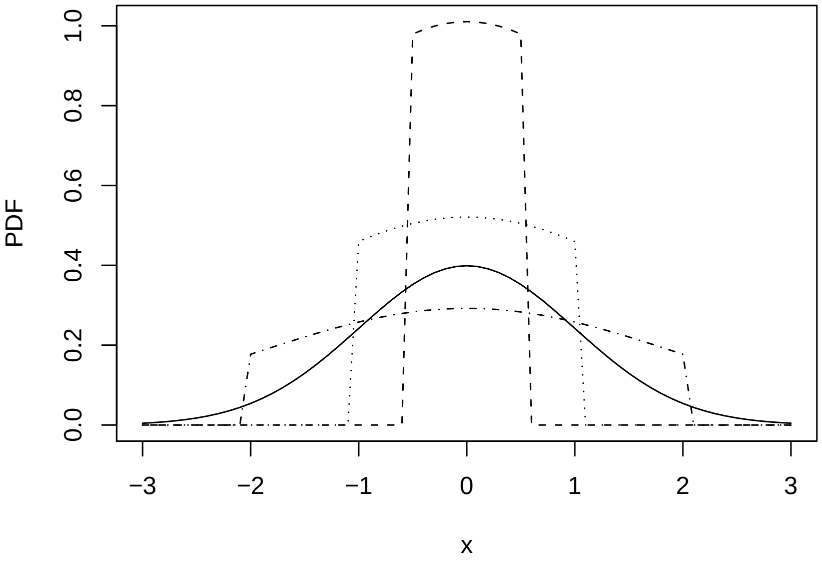

Figure 1: Truncated pdfs of the standard normal distribution.

4. Example 2

This example draws the quantile–quantile plot for simulated data from a truncated Weibull distribution. We considered the distribution given by the cdf

$$F(x) \quad = \quad 1 - \exp\left(-x^2\right) \tag{7}$$

(for $x > 0$) truncated at $a = 1$ and $b = 2$. The functions `rtrunc()` and `qtrunc()` from Table 1 are used. The argument `spec` is taken to be `"weibull"`.

For a simulated data set of size 100, the plot of the expected order statistics versus the observed order statistics is shown in Figure 2.

```
R> set.seed(1)
R> x <- rtrunc(100, "weibull", a = 1, b = 2, shape = 2)
R> x <- sort(x)
R> y <- qtrunc((seq(1, 100) - 0.375)/(100 + 0.25), "weibull", a = 1,
+     b = 2, shape = 2)
R> m1 <- min(x, y)
R> m2 <- max(x, y)
R> plot(x, y, xlim = c(m1, m2), ylim = c(m1, m2), xlab = "Observed",
+     ylab = "Expected")
R> abline(0, 1)
```

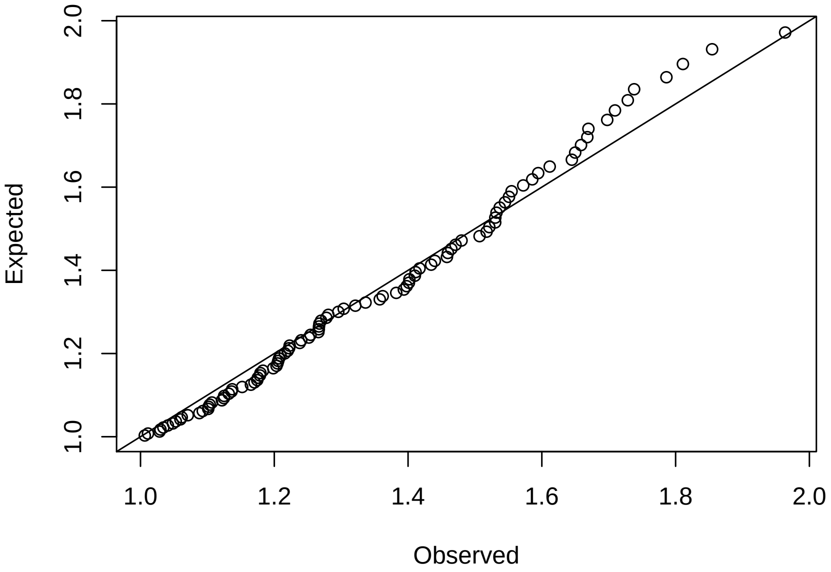

Figure 2: Quantile–quantile plot for the simulated sample from (7).

5. Conclusions

We have provided programs in R (a freely available statistical software) for computing quantities of interest—six of them—for truncated distributions. The computed quantities include: the probability density function, mean, variance, cumulative distribution function, quantile function and n random numbers. These programs could have wide applicability because: 1) no restrictions are imposed on the input parameters n, a, b, g and G; 2) six quantities of interest are given; 3) the programs simple and easy to implement on any platform; and 4) full control is given to the user as far as accuracy and stability of the results.

Acknowledgments

The authors would like to thank the Editor-in-Chief and the Associate Editors for carefully reading the paper and for their comments which greatly improved the paper.

References

Bhowmick K, Mukhopadhyay A, Mitra GB (2000). "Edgeworth Series Expansion of the Truncated Cauchy Function and its Effectiveness in the Study of Atomic Heterogeneity." *Zeitschrift für Kristallographie*, **215**, 718–726.

Cho BR, Govindaluri MS (2002). "Optimal Screening Limits in Multi-Stage Assemblies." *International Journal Production Research*, **40**, 1993–2009.

Field T, Harder U, Harrison P (2004). "Network Traffic Behaviour in Switched Ethernet Systems." *Performance Evaluation*, **58**, 243–260.

Ihaka R, Gentleman R (1996). "R: A Language for Data Analysis and Graphics." *Journal of Computational and Graphical Statistics*, **5**, 299–314.

Jeang A (1997). "An Approach of Tolerance Design for Quality Improvement and Cost Reduction." *International Journal Production Research*, **35**, 1193–1211.

Kapur KC, Cho BR (1994). "Economic Design and Development of Specification." *Quality Engineering*, **6**, 401–417.

Kapur KC, Cho BR (1996). "Economic Design of the Specification Region for Multiple Quality Characteristics." *IIE Transactions*, **28**, 237–248.

Khasawneh MT, Bowling SR, Kaewkuekool S, Cho BR (2004). "Tables of a Truncated Standard Normal Distribution: A Singly Truncated Case." *Quality Engineering*, **17**, 33–50.

Khasawneh MT, Bowling SR, Kaewkuekool S, Cho BR (2005). "Tables of a Truncated Standard Normal Distribution: A Doubly Truncated Case." *Quality Engineering*, **18**, 227–241.

Phillips MD, Cho BR (1998). "Quality Improvement for Processes with Circular and Spherical Specification Region." *Quality Engineering*, **11**, 235–243.

Phillips MD, Cho BR (2000). "Modeling of Optimum Specification Regions." *Applied Mathematical Modelling*, **24**, 327–341.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Shmueli U (1979). "Symmetry and Composition Dependent Cumulative Distribution of the Normalized Structure Amplitude for Use in Intensity Statistics." *Acta Crystallography A*, **35**, 282–286.

Shmueli U, Wilson AJC (1981). "Effects of Space Group Symmetry and Atomic Heterogeneity on Intensity Statistics." *Acta Crystallography A*, **37**, 342–353.

Affiliation:

Saralees Nadarajah
School of Mathematics
University of Manchester
Manchester M13 9PL, United Kingdom
E-mail:saralees.nadarajah@manchester.ac.uk

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 16, Code Snippet 2 *Submitted:* 2006-06-06
August 2006 *Accepted:* 2006-08-17