DIGITALCOMMONS @WAYNESTATE

Journal of Modern Applied Statistical Methods

Volume 4 | Issue 1 Article 27

5-1-2005

JMASM16: Pseudo-Random Number Generation In R For Some Univariate Distributions

Hakan Demirtas

University of Illinois at Chicago, demirtas@uic.edu

Journal of Modern Applied Statistical Methods
May, 2005, Vol. 3, No. 1, 300-311

1538 – 9472/05/$95.00

JMASM Algorithms and Code

JMASM16: Pseudo-Random Number Generation In R For Some Univariate Distributions

Hakan Demirtas
School of Public Health
University of Illinois at Chicago

An increasing number of practitioners and applied researchers started using the R programming system in recent years for their computing and data analysis needs. As far as pseudo-random number generation is concerned, the built-in generator in R does not contain some important univariate distributions. In this article, complementary R routines that could potentially be useful for simulation and computation purposes are provided.

Key words: Simulation; computation; pseudo-random numbers

Introduction

Following upon the work of Demirtas (2004), pseudo-random generation functions written in R for some univariate distributions are presented. The built-in pseudo-random number generator in R does not have routines for some important univariate distributions. Built-in codes are available only for the following univariate distributions: uniform, normal, chi-square, *t*, F, lognormal, exponential, gamma, Weibull, Cauchy, beta, logistic, stable, binomial, negative binomial, Poisson, geometric, hypergeometric and Wilcoxon.

The purpose of this article is to provide complementary R routines for generating pseudo-random numbers from some univariate distributions. In the next section, eighteen R functions of which the first thirteen correspond to the distributions that are not contained in the generator (Codes 1-13) are presented. The quality of the resulting variates have not been tested in the computer science sense. However, the first three moments for each distribution were rigorously tested. For the purposes of most applications, fulfillment of this criterion should be a reasonable approximation to reality. The last 5 functions (Codes 14-18) address already available univariate distributions; the reason for their inclusion is that variates generated with these routines are of a slightly better quality than those generated by the built-in code in terms of above-mentioned criterion.

Hakan Demirtas is an Assistant Professor of Biostatistics at the University of Illinois at Chicago. His research interests are the analysis of incomplete longitudinal data, multiple imputation and Bayesian computing. E-mail address: demirtas@uic.edu.

Functions for random number generation

The following abbreviations are used: *PDF* stands for the probability density function; *PMF* stands for the probability mass function; *CDF* stands for the cumulative distribution function; *GA* stands for the generation algorithm and *EAA* stands for an example of application areas; *nrep* stands for the number of identically and independently distributed random variates. The formal arguments other than *nrep* reflect the parameters in *PDF* or *PMF*. $E(X)$ and $V(X)$ denote the expectation and the variance of the random variable X, respectively.

Left truncated normal distribution

PDF: $$f(x \mid \mu,\sigma,\tau) = \frac{e^{-(x-\mu)^2/(2\sigma^2)}}{\sqrt{2\pi}\sigma(1-\Phi(\frac{\tau-\mu}{\sigma}))}$$

for $\tau \le x < \infty$ where $\Phi()$ is the standard normal *CDF*, μ, σ and τ are the mean, standard deviation and left truncation point, respectively. *EAA*: Modeling the tail behavior in simulation studies. *GA*: Robert's (1995) acceptance/ rejection algorithm with a shifted exponential as the majorizing density. For $\mu=0$ and $\sigma=1$, $E(X) = \frac{e^{-\tau^2/2}}{\sqrt{2\pi}(1-\Phi(\tau))}$, $V(X)$ is a complicated function of τ (see Code 1).

Left truncated gamma distribution

PDF:

$$f(x \mid \alpha,\beta) = \frac{1}{(\Gamma(\alpha)-\Gamma_{\tau/\beta}(\alpha))\beta^{\alpha}} x^{\alpha-1}e^{-x/\beta}$$

for $\tau \le x < \infty$, $\alpha>1$ and $min(\tau,\beta)>0$ where α and β are the shape and scale parameters, respectively, τ is the cutoff point at which truncation occurs and $\Gamma_{\tau/\beta}$ is the incomplete gamma function. *EAA*: Modeling left-censored data. *GA*: An acceptance/rejection algorithm (Dagpunar, 1978) where the majorizing density is chosen to be a truncated exponential.

$$E(X) = \beta[\frac{\Gamma(\alpha+1)-\Gamma_{\tau}(\alpha+1)}{\Gamma(\alpha)-\Gamma_{\tau}(\alpha)}],$$

$$V(X) = \beta^2[\frac{\Gamma(\alpha+2)-\Gamma_{\tau}(\alpha+2)}{\Gamma(\alpha)-\Gamma_{\tau}(\alpha)}] - E(X)^2.$$

The procedure works best when τ is small (see Code 2).

Code 1. Left truncated normal distribution:

```
draw.left.truncated.normal<-function(nrep,mu,sigma,tau){
if (sigma<=0){
stop("Standard deviation must be positive!\n")}
lambda.star<-(tau+sqrt(tau^2+4))/2
accept<-numeric(nrep) ; for (i in 1:nrep){
sumw<-0 ; while (sumw<1){
y<-rexp(1,lambda.star)+tau
gy<-lambda.star*exp(lambda.star*tau)*exp(-lambda.star*y)
fx<-exp(-(y-mu)^2/(2*sigma^2))/(sqrt(2*pi)*sigma*(1-pnorm((tau-
mu)/sigma)))
ratio1<-fx/gy ; ratio<-ratio1/max(ratio1)
u<-runif(1); w<-(u<=ratio) ; accept[i]<-y[w]; sumw<-sum(w)}}
accept}
```

Code 2: Left truncated gamma distribution

```
draw.left.truncated.gamma<-function(nrep,alpha,beta,tau){
if (tau<0){stop("Cutoff point must be positive!\n")}
if ((alpha<=1)){stop("Shape parameter must be greater than 1!\n")}
if ((beta<=0)){stop("Scale parameter must be positive!\n")}
y<-numeric(nrep); for (i in 1:nrep){
index<-0 ; scaled.tau<-tau/beta
lambda<-(scaled.tau-alpha+sqrt((scaled.tau-
alpha)^2+4*scaled.tau))/(2*scaled.tau)
while (index<1){
u<-runif(1); u1<-runif(1) ; y[i]<-(-log(u1)/lambda)+tau
w<-((1-lambda)*y[i]-(alpha-1)*(1+log(y[i])+log((1-lambda)/(alpha-
1)))<=-log(u))
index<-sum(w)}} ; y<-y*beta
y}
```

Laplace (double exponential) distribution

PDF: $f(x)=\frac{\lambda}{2}e^{-\lambda|x-\alpha|}$ for $\lambda>0$, where α and λ are the location and scale parameters, respectively. *EAA*: Monte Carlo studies of robust procedures, because it has a heavier tail than the normal distribution. *GA*: A sample from an exponential distribution with mean λ is generated, then the sign is changed with 1/2 probability and the resulting variates get shifted by α. $E(X)=\alpha$, $V(X)=2/\lambda^2$ (see Code 3).

Inverse Gaussian distribution

PDF:

$$f(x\mid\mu,\lambda)=\left(\frac{\lambda}{2\pi}\right)^{1/2}x^{-3/2}e^{-\frac{\lambda(x-\mu)^2}{2\mu^2x}}$$ for $x\geq 0$, $\mu>0$, $\lambda>0$, where μ and λ are the location and scale parameters, respectively. *EAA*: Reliability studies. *GA*: An acceptance/rejection algorithm developed by Michael et al. (1976). $E(X)=\mu$, $V(X)=\mu^3/\lambda$ (see Code 4).

Von Mises distribution

PDF: $f(x\mid K)=\frac{1}{2\pi I_0(K)}e^{K\cos(x)}$ for $-\pi\leq x\leq\pi$ and $K>0$, where $I_0(K)$ is a modified Bessel function of the first kind of order 0. *EAA*: Modeling directional data. *GA*: Acceptance/rejection method of Best and Fisher (1979) that uses a transformed folded Cauchy distribution as the majorizing density. $E(X)=0$ (see Code 5).

Zeta (Zipf) distribution

PDF: $f(x\mid\alpha)=\frac{1}{\zeta(\alpha)x^{\alpha}}$ for $x=1,2,3,\ldots$ and $\alpha>1$, where $\zeta(\alpha)=\sum_{x=1}^{\infty}x^{-\alpha}$ (Riemann zeta function). *EAA*: Modeling the frequency of random processes. *GA*: Acceptance/rejection algorithm of Devroye (1986).

$$E(X)=\frac{\zeta(\alpha-1)}{\zeta(\alpha)},$$

$$V(X)=\frac{\zeta(a)\zeta(a-2)-(\zeta(a-1))^2}{(\zeta(a))^2},$$

(see Code 6).

Code 3. Laplace (double exponential) distribution:

```
draw.laplace<-function(nrep, alpha, lambda){
if (lambda<=0){stop("Scale parameter must be positive!\n")}
y<-rexp(nrep,lambda)
change.sign<-sample(c(0,1), nrep, replace = TRUE)
y[change.sign==0]<--y[change.sign==0] ; laplace<-y+alpha
laplace}
```

Code 4. Inverse Gaussian distribution:

```
draw.inverse.gaussian<-function(nrep,mu,lambda){
if (mu<=0){stop("Location parameter must be positive!\n")}
if (lambda<=0){stop("Scale parameter must be positive!\n")}
inv.gaus<-numeric(nrep); for (i in 1:nrep){
v<-rnorm(1) ; y<-v^2
x1<-mu+(mu^2*y/(2*lambda))-(mu/(2*lambda))*(sqrt(4*mu*lambda*y+mu^2*y^2))
u<-runif(1) ; inv.gaus[i]<-x1
w<-(u>(mu/(mu+x1))) ; inv.gaus[i][w]<-mu^2/x1}
inv.gaus}
```

Code 5. Von Mises distribution:

```
draw.von.mises<-function(nrep,K){
if (K<=0){stop("K must be positive!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
index<-0 ; while (index<1){
u1<-runif(1) ; u2<-runif(1); u3<-runif(1)
tau<-1+(1+4*K^2)^0.5 ; rho<-(tau-(2*tau)^0.5)/(2*K)
r<-(1+rho^2)/(2*rho) ; z<-cos(pi*u1)
f<-(1+r*z)/(r+z) ; c<-K*(r-f)
w1<-(c*(2-c)-u2>0) ; w2<-(log(c/u2)+1-c>=0)
y<-sign(u3-0.5)*acos(f) ; x[i][w1|w2]<-y
index<-1*(w1|w2)}}
x}
```

Code 6. Zeta (Zipf) distribution

```
draw.zeta<-function(nrep,alpha){
if (alpha<=1){stop("alpha must be greater than 1!\n")}
zeta<-numeric(nrep) ; for (i in 1:nrep){
index<-0 ; while (index<1){
u1<-runif(1) ; u2<-runif(1)
x<-floor(u1^(-1/(alpha-1))) ; t<-(1+1/x)^(alpha-1)
w<-x<(t/(t-1))*(2^(alpha-1)-1)/(2^(alpha-1)*u2)
zeta[i]<-x ; index<-sum(w)}}
zeta}
```

Logarithmic distribution

PMF: $f(x|\theta) = -\frac{\theta^x}{x log(1-\theta)}$ for x=1,2,3,... and 0<θ<1. *EAA*: Modeling the number of items processed in a given period of time. *GA*: The chop-down search method of Kemp (1981). $E(X) = \frac{-\theta}{(1-\theta)} \frac{1}{log(1-\theta)}$, $V(X) = \frac{-\theta log(1-\theta) - \theta^2}{(1-\theta)^2 (log(1-\theta))^2}$ (see Code 7).

Beta-binomial distribution

PMF: $f(x|n,\alpha,\beta) = \frac{n!}{x!(n-x)!B(\alpha,\beta)} \int_0^1 \pi^{\alpha-1+x} (1-\pi)^{n+\beta-1-x} d\pi$ for x=0,1,2,..., α>0 and β>0, where n is the sample size, α and β are the shape parameters and B(α,β) is the complete beta function. *EAA*: Modeling overdispersion or extravariation in applications where clusters of separate binomial distributions. *GA*: First π is generated as the appropriate beta and then it is used as the success probability in binomial. $E(X) = \frac{n\alpha}{\alpha+\beta}$, $V(X) = \frac{n\alpha\beta(\alpha+\beta+n)}{(\alpha+\beta)^2(\alpha+\beta+1)}$ (see Code 8).

Code 7. Logarithmic distribution:

```
draw.logarithmic<-function(nrep,theta){
if ((theta<=0)|(theta>1)){stop("theta must be between 0 and 1!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
index<-0 ; x0<-1 ; u<-runif(1)
while (index<1){t<--(theta^x0)/(x0*log(1-theta))
px<-t ; w<-(u<=px) ; x[i]<-x0 ; u<-u-px
index<-sum(w) ; x0<-x0+1}}
x}
```

Code 8. Beta-binomial distribution:

```
draw.beta.binomial<-function(nrep,alpha,beta,n){
if ((alpha<=0)|(beta<=0)){stop("alpha and beta must be positive!\n")}
if (floor(n)!=n){stop("Size must be an integer!\n")}
if (floor(n)<2){stop("Size must be greater than 2!\n")}
beta.variates<-numeric(nrep) ; beta.binom<-numeric(nrep)
for (i in 1:nrep){
beta.variates[i]<-rbeta(1,alpha,beta)
beta.binom[i]<-rbinom(1,n,beta.variates[i])}
beta.binom}
```

Rayleigh distribution

$$PDF: \quad f(x\,|\,\sigma)=\frac{x}{\sigma^2}e^{-x^2/2\sigma^2} \quad \text{for } x\geq 0$$

and $\sigma>0$, where σ is the scale parameter. *EAA*: Modeling spatial patterns. *GA*: The inverse *CDF* method. $E(X)=\sigma\sqrt{\pi/2}$, $V(X)=\sigma^2(4-\pi)/2$ (see Code 9).

Pareto distribution

$$PDF: \quad f(x|a,b)=\frac{ab^a}{x^{a+1}} \quad \text{for}$$

$0<b\leq x<\infty$ and $a>0$, where a and b are the shape and location parameters, respectively. *EAA*: Gene filtering in microarray experiments. *GA*: The inverse *CDF* method. $E(X)=\frac{ab}{a-1}$, $V(X)=\frac{ab^2}{(a-2)(a-1)^2}$. The procedure works best when a and b are not too small (see Code 10).

Non-central *t* distribution

Describes the ratio $\frac{Y}{\sqrt{U/\nu}}$ where U is a central chi-square random variable with ν degrees of freedom and Y is an independent normally distributed random variable with variance 1 and mean λ. *EAA*: Thermodynamic stability scores. *GA*: Based on arithmetic functions of normal and χ^2 variates. $E(X)=\lambda\sqrt{\nu/2}\,\frac{\Gamma((\nu-1)/2)}{\Gamma(\nu/2)}$, $V(X)=(1+\lambda^2)\nu-\mathrm{E}(X^2)$ (see Code 11).

Code 9. Rayleigh distribution:

```
draw.rayleigh<-function(nrep,sigma){
if (sigma<=0){stop("Standard deviation must be positive!\n")}
u<-runif(nrep); rayl<-sigma*sqrt(-2*log(u))
rayl}
```

Code 10: Pareto distribution:

```
draw.pareto<-function(nrep,shape,location){
if (shape<=0){stop("Shape parameter must be positive!\n")}
if (location<=0){stop("Location parameter must be positive!\n")}
u<-runif(nrep) ; pareto<-location/(u^(1/shape))
pareto}
```

Code 11. Non-central *t* distribution:

```
draw.noncentral.t<-function(nrep,nu,lambda){
if (nu<=1){stop("Degrees of freedom must be greater than 1!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
x[i]<-rt(1,nu)+(lambda/sqrt(rchisq(1,nu)/nu))}
x}
```

Non-central chi-squared distribution

PDF:

$$f(x\mid\lambda,\nu)=\frac{e^{-(x+\lambda)/2}x^{\nu/2-1}}{2^{\nu/2}}\sum_{k=0}^{\infty}\frac{(\lambda x)^k}{4^k\,k!\,\Gamma(k+\nu/2)}$$

for $0\le x\le\infty$, $\lambda>0$ and $\nu>1$, where λ is the non-centrality parameter and ν is degrees of freedom. Both λ and ν can be non-integers. *EAA*: Wavelets in biomedical imaging. *GA*: Based on the sum of squared standard normal deviates. $E(X)=\lambda+\nu$, $V(X)=4\lambda+2\nu$ (see Code 12).

Doubly non-central *F* distribution

Describes the ratio of two scaled non-central χ^2 variables; that is, $F=\frac{X_1^2/n}{X_2^2/m}$ for $X_1^2\sim\chi^2(n,\lambda_1)$ and $X_2^2\sim\chi^2(m,\lambda_2)$, where n and m are numerator and denumerator degrees of freedom, respectively; λ_1 and λ_2 are the numerator and denumerator non-centrality parameters, respectively. *EAA*: Biomedical microarray studies. *GA*: Simple ratio of non-central χ^2 variables adjusted by corresponding degrees of freedom. $E(X)$ and $V(X)$ are too complicated to include here (see Code 13).

Code 12. Non-central chi-squared distribution:

```
draw.noncentral.chisquared<-function(nrep,df,ncp){
if (ncp<0){stop("Non-Centrality parameter must be non-negative!\n")}
if (df<=1){stop("Degrees of freedom must be greater than 1!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
df.int<-floor(df) ; df.frac<-df-df.int
mui<-sqrt(ncp/df.int) ; jitter<-0
if (df.frac!=0){jitter<-rchisq(1,df.frac)}
x[i]<-sum((rnorm(df.int)+mui)^2)+jitter}
x}
```

Code 13. Doubly non-central F distribution:

```
draw.noncentral.F<-function(nrep,df1,df2,ncp1,ncp2){
if (ncp1<0){stop("Numerator non-centrality parameter must be non-
negative!\n")}
if (ncp2<0){stop("Denominator non-centrality parameter must be non-
negative!\n")}
if (df1<=1){stop("Numerator degrees of freedom must be greater than
1!\n")}
if (df2<=1){
stop("Denominator degrees of freedom must be greater than 1!\n")}
x<-draw.noncentral.chisquared(nrep,df1,ncp1)/
draw.noncentral.chisquared(nrep,df2,ncp2)
x}
```

Standard *t* distribution

PDF:

$$f(x\mid\nu)=\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi}}(1+\frac{x^2}{\nu})^{-(\nu+1)/2} \text{ for}$$

$-\infty<x<\infty$, where ν is the degrees of freedom and Γ() is the complete gamma function. *GA*: A rejection polar method developed by Bailey (1994). $E(X)=0$, $V(X)=\frac{\nu}{\nu-2}$, (see Code 14).

Weibull distribution

PDF: $f(x\mid\alpha,\beta)=\frac{\alpha}{\beta^{\alpha}}x^{\alpha-1}e^{-(x/\beta)^{\alpha}}$ for $0\le x<\infty$ and $min(\alpha,\beta)>0$, where α and β are the shape and scale parameters, respectively. *EAA*: Modeling lifetime data. *GA*: The inverse *CDF* method.

$E(X)=(1+1/\alpha)\beta$,

$V(X)=\left[\Gamma(1+2/\alpha)-\Gamma^2(1+1/\alpha)\right]\beta^2$,

(see Code 15).

Code 14. Standard **t** distribution:

```
draw.t<-function(nrep,df){
if (df<=1){stop("Degrees of freedom must be greater than 1!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
index<-0 ; while (index<1){
v1<-runif(1,-1,1) ; v2<-runif(1,-1,1); r2<-v1^2+v2^2
r<-sqrt(r2) ; w<-(r2<1)
x[i]<-v1*sqrt(abs((df*(r^(-4/df)-1)/r2)))
index<-sum(w)}}
x}
```

Code 15. Weibull distribution:

```
draw.weibull<-function(nrep, alpha, beta){
if ((alpha<=0)|(beta<=0)){
stop("alpha and beta must be positive!\n")}
u<-runif(nrep) ; weibull<-beta*((-log(u))^(1/alpha))
weibull}
```

Gamma distribution when $\alpha<1$

$$PDF: \quad f(x|\alpha,\beta)=\frac{1}{\Gamma(\alpha)\beta^{\alpha}}x^{\alpha-1}e^{-x/\beta}$$

for $0\le x<\infty$, $min(\alpha,\beta)>0$, where α and β are the shape and scale parameters, respectively. *EAA*: Bioinformatics. *GA*: An acceptance/rejection algorithm developed by Ahrens and Dieter (1974) and Best (1983). It works when $\alpha<1$. $E(X)=\alpha\beta$, $V(X)=\alpha\beta^2$ (see Code 16).

Gamma distribution when $\alpha>1$

PDF: Same as before. *EAA*: Bioinformatics. *GA*: A ratio of uniforms method introduced by Cheng and Feast (1979). It works when $\alpha>1$. $E(X)=\alpha\beta$, $V(X)=\alpha\beta^2$ (see Code 17).

Code 16. Gamma distribution when $\alpha<1$

```
draw.gamma.alpha.less.than.one<-function(nrep,alpha,beta){
if (beta<=0){stop("Scale parameter must be positive!\n")}
if ((alpha<=0)|(alpha>=1)){
stop("Shape parameter must be between 0 and 1!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
index<-0 ; while (index<1){
u1<-runif(1) ; u2<-runif(1)
t<-0.07+0.75*sqrt(1-alpha) ; b<-1+exp(-t)*alpha/t
v<-b*u1 ; w1<-(v<=1) ; w2<-(v>1)
x1<-t*(v^(1/alpha)) ; w11<-(u2<=(2-x1)/(2+x1))
w12<-(u2<=exp(-x1)) ; x[i][w1&w11]<-x1[w1&w11]
x[i][w1&!w11&w12]<-x1[w1&!w11&w12]
x2=-log(t*(b-v)/alpha) ; y<-x2/t
w21<-(u2*(alpha+y*(1-alpha))<=1)
w22<-(u2<=y^(alpha-1)) ; x[i][w2&w21]<-x2[w2&w21]
x[i][w2&!w21&w22]<-x2[w2&!w21&w22]
index<-1*(w1&w11)+1*(w1&!w11&w12)+1*(w2&w21)+1*(w2&!w21&w22)}}
x<-beta*x
x}
```

Code 17. Gamma distribution when $\alpha>1$:

```
draw.gamma.alpha.greater.than.one<-function(nrep,alpha,beta){
if (beta<=0){stop("Scale parameter must be positive!\n")}
if (alpha<=1){stop("Shape parameter must be greater than 1!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
index<-0 ; while (index<1){
u1<-runif(1); u2<-runif(1)
v<-(alpha-1/(6*alpha))*u1/((alpha-1)*u2)
w1<-((2*(u2-1)/(alpha-1))+v+(1/v)<=2)
w2<-((2*log(u2)/(alpha-1))-log(v)+v<=1)
x[i][w1]<-(alpha-1)*v ; x[i][!w1&w2]<-(alpha-1)*v
index<-1*w1+1*(!w1&w2)}}
x<-x*beta
x}
```

Beta distribution when *max*(α,β)<1

PDF:

$$f(x \mid \alpha, \beta) = \frac{1}{B(\alpha, \beta)} x^{\alpha-1}(1-x)^{\beta-1}$$ for

$0 \le x \le 1$, $0 \le \alpha < 1$ and $0 \le \beta < 1$, where α and β are the shape parameters and $B(\alpha,\beta)$ is the complete beta function. *EAA*: Analysis of biomedical signals. *GA*: An acceptance/rejection algorithm developed by Johnk (1964). It works when both parameters are less than 1. $E(X)=\frac{\alpha}{\alpha+\beta}$,

$V(X) = \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}$ (see Code 18).

found to be negligible, suggesting that random number generation routines presented are accurate. These routines could be a handy addition to a practitioner's set of tools given the growing interest in R. However, the reader is invited to be cautious about the following issues: 1) It is not postulated that algorithms presented are the most efficient. Furthermore, implementation of a given algorithm may not be optimal. Given sufficient time and resources, one can write more efficient routines. 2) Quality of every random number generation process depends on the uniform number generator.

Code 18. Beta distribution when *max*(α,β)<1:

```
draw.beta.alphabeta.less.than.one<-function(nrep,alpha,beta){
if ((alpha>=1)|(alpha<=0)|(beta>=1)|(beta<=0)) {
stop ("Both shape parameters must be between 0 and 1!\n")}
x<-numeric(nrep) ; for (i in 1:nrep){
index<-0 ; while (index<1){
u1<-runif(1) ; u2<-runif(1)
v1<-u1^(1/alpha) ; v2<-u2^(1/beta)
summ<-v1+v2 ; w<-(summ<=1)
x[i]<-v1/summ ; index<-sum(w)}}
x}
```

Results for arbitrarily chosen parameter values

For each distribution, the parameters can take infinitely many values and first two moments virtually fluctuate on the entire real line. The quality of random variates was tested by a broad range of simulations to see any potential aberrances and abnormalities in some subset of the parameter domains and to avoid any selection biases. The empirical and theoretical moments for arbitrarily chosen parameter values are reported in Table 1 and 2.

Table 1 tabulates the theoretical and empirical means for each distribution for arbitrary values. Throughout the table, the number of replications (*nrep*) is chosen to be 10,000. A similar comparison is made for the variances, as shown in Table 2. In both tables, the deviations from the expected moments are

McCullough (1999) raised some questions about the quality of Splus generator. At the time of this writing, a source that tested the R generator is unknown to the author. In addition, the differences between empirical and distributional moments have merely been examined for each distribution. More comprehensive and computer science-minded tests are needed possibly using DIEHARD suite (Marsaglia, 1995) or other well-regarded test suites.

Table 1: Comparison of theoretical and empirical means for arbitrarily chosen parameter values.

Distribution	*Parameter(s)*	*Theoretical mean*	*Empirical mean*
Left truncated normal	$\mu=0$, $\sigma=1$, $\tau=0.5$	1.141078	1.143811
Left truncated gamma	$\alpha=4$, $\beta=2$, $\tau=0.5$	8.002279	8.005993
Laplace	$\alpha=4$, $\lambda=2$	4	3.999658
Inverse Gaussian	$\mu=1$, $\lambda=1$	1	1.001874
Von Mises	$K=10$	0	0.002232
Zeta (Zipf)	$\alpha=4$	1.110626	1.109341
Logarithmic	$\theta=0.6$	1.637035	1.637142
Beta-binomial	$\alpha=2$, $\beta=3$, $n=10$	4	4.016863
Rayleigh	$\sigma=4$	5.013257	5.018006
Pareto	$a=5$, $b=5$	6.25	6.248316
Non-central t	$\nu=5$, $\lambda=1$	1.189416	1.191058
Non-central Chi-squared	$\nu=5$, $\lambda=2$	7	7.004277
Doubly non-central F	$n=5$, $m=10$, $\lambda_1=2$, $\lambda_2=3$	0.667381	0.666293
Standard t	$\nu=5$	0	0.001263
Weibull	$\alpha=5$, $\beta=5$	4.590844	4.587294
Gamma with $\alpha<1$	$\alpha=0.3$, $\beta=0.4$	0.12	0.118875
Gamma with $\alpha>1$	$\alpha=3$, $\beta=0.4$	1.2	1.200645
Beta with $\alpha<1$ and $\beta<1$	$\alpha=0.7$, $\beta=0.4$	0.636363	0.636384

Table 2: Comparison of theoretical and empirical variances for arbitrarily chosen parameter values.

Distribution	*Parameter(s)*	*Theoretical variance*	*Empirical variance*
Left truncated normal	$\mu=0$, $\sigma=1$, $\tau=0.5$	0.603826	0.602914
Left truncated gamma	$\alpha=4$, $\beta=2$, $\tau=0.5$	15.98689	15.86869
Laplace	$\alpha=4$, $\lambda=2$	0.5	0.502019
Inverse Gaussian	$\mu=1$, $\lambda=1$	1	0.997419
Zeta (Zipf)	$\alpha=4$	0.545778	0.556655
Logarithmic	$\theta=0.6$	1.412704	1.4131545
Beta-binomial	$\alpha=2$, $\beta=3$, $n=10$	6	6.001696
Rayleigh	$\sigma=4$	6.867259	6.854438
Pareto	$a=5$, $b=5$	2.604167	2.604605
Non-central t	$\nu=5$, $\lambda=1$	1.918623	1.903359
Non-central Chi-squared	$\nu=5$, $\lambda=2$	18	18.09787
Doubly non-central F	$n=5$, $m=10$, $\lambda_1=2$, $\lambda_2=3$	0.348817	0.346233
Standard t	$\nu=5$	1.666667	1.661135
Weibull	$\alpha=5$, $\beta=5$	1.105749	1.098443
Gamma with $\alpha<1$	$\alpha=0.3$, $\beta=0.4$	0.048	0.047921
Gamma with $\alpha>1$	$\alpha=3$, $\beta=0.4$	0.48	0.481972
Beta with $\alpha<1$ and $\beta<1$	$\alpha=0.7$, $\beta=0.4$	0.110193	0.110126

References

Ahrens, J. H., & Dieter, U. (1974). Computer methods for sampling from gamma, beta, poisson and binomial distributions. Computing, 1, 223-246.

Bailey, R. W. (1994). Polar generation of random variates with the t-distribution. *Mathematics of Computation, 62*, 779-781.

Best, A. W. (1983). A note on gamma variate generators with shape parameter less than unity. *Computing, 30*, 185-188.

Best, D. J., & Fisher, N. I. (1979). Efficient simulation of the von mises distribution. *Applied Statistics, 28*, 152-157.

Cheng, R. C. H., & Feast, G. M. (1979). *Some simple gamma variate generation*. *Applied Statistics, 28*, 290-295.

Dagpunar, J. S. (1978). Sampling of variates from a truncated gamma distribution. *Journal of Statistical Computation and Simulation, 8*, 59-64.

Demirtas, H. (2004). Pseudo-random number generation in r for commonly used multivariate distributions. *Journal of Modern Applied Statistical Methods*, 3, 385-497.

Devroye, L. (1986). *Non-Uniform random variate generation*. New York: Springer-Verlag.

Jhonk, M. D. (1964). Erzeugung von betaverteilter und gammaverteilter zufallszahlen. *Metrika, 8*, 5-15.

Kemp, A. W. Efficient generation of logarithmically distributed pseudo-random variables. *Applied Statistics, 30*, 249-253.

Marsaglia, G. (1995). *The marsaglia random number cdrom, including the diehard battery of tests of randomness*. Department of Statistics, Florida State University, Tallahassee, Florida.

McCullough, B. D. (1999). Assessing the reliability of statistical software: Part 2. *The American Statistician, 53*, 149-159.

Michael, J. R., William, R. S., & Haas, R. W. (1976). Generating random variates using transformations with multiple roots. The American Statistician, 30, 88-90.

Robert, C. P. (1995). Simulation of truncated random variables. *Statistics and Computing, 13*, 169-184.