Chapter 14: Distributions

Scilab contains numerous functions pertaining to statistical distributions (see in scilab manual chapter 2.12: Cumulative Distribution Functions, Inverses, Random variables): for the usual statistical laws (beta, binomial, normal, chi-square, Fisher, gamma, Poisson, Student) scilab provides the cumulative distribution function, the p-value or the number of degrees of freedom, given the other parameters: for instance, [P,Q]=cdfchi("PQ",X,Df) provides the p-value of the chi-square distribution with X degrees of freedom at value X ; [X]=cdfchi("X",Df,P,1-P) gives the number that a chi-square distribution has a probability P to be greater than ; [Df]=cdfchi("Df",P,1-PX) gives the number of degrees of freedom of the chi-square whose probability of being greater than X is P (see scilab manual for more details). Scilab provides also random number generators for the uniform and the normal laws.

Scilab does not however provide random number generators for the other laws, nor does it include some useful laws, such as: the hypergeometric, the lognormal and the logistic laws. So, GROCER includes for these three laws functions which provide their cumulative distribution function, their probability distribution function, their inverse function and which generate random numbers taken from these distributions. GROCER provides also probability distribution function and random generators for all distribution for which they were lacking in scilab: beta, binomial, normal (probability distribution function), chi-square, Fisher, gamma, Poisson, Student.

These functions are mostly straightforward translations of James LeSage distribution functions and the interested reader is invited to read the corresponding chapter in James LeSage (1999) book. Accordingly, the functions are not conformable to the existing distribution functions in Scilab. I do not think it could be a problem, but, if it were, I can consider fixing it…

beta_____ **beta function**

CALLING SEQUENCE

[x]=beta(a,b)

PARAMETERS

INPUT:
* a = a vector
* b = a vector with the same size

OUTPUT:
x=beta(a,b)

DESCRIPTION
computes the beta function for vectors a and b

Example:
y=beta([3 5 6],[2 7 3])

beta_pdf_____beta probability distribution

CALLING SEQUENCE

[pdf]=beta_pdf(x,a,b)

PARAMETERS

INPUT:
* x = vector of components
* a = beta distribution parameter, a = scalar or vector of the same size than x
* b = beta distribution parameter b = scalar or vector of the same size than x
---
OUTPUT: probability distribution function at each element of x of the beta(a,b) distribution

DESCRIPTION
Probability distribution function of the beta(a,b) distribution.
NOTE: mean[(beta(a,b)] = a/(a+b), variance = ab/((a+b)*(a+b)*(a+b+1))

Example:
y=beta(x,10,5)

beta_rnd_____beta random draws

CALLING SEQUENCE

[rnd]=beta_rnd(n,a,b)

PARAMETERS

INPUT:
* n = size of the vector of draws
* a = beta distribution parameter, a = scalar
* b = beta distribution parameter b = scalar

OUTPUT:
t = n-vector of random draws from the beta(a,b) distribution

DESCRIPTION
Random draws from the beta(a,b) distribution.

Example:
x=beta_rnd(100,10,5)

bincoef_____binomial coefficients

CALLING SEQUENCE

[k]=bincoef(n,N)

PARAMETERS

INPUT:
* n = (mxp) matrix
* N = (nxp) matrix (same size than n) or scalar

OUTPUT:
* k = (mxp) matrix of binomial coefficents with
 - k(i,j)=binomial(n(i,j),N(i,j)) or
 - k(i,j)=binomial(n(i,j),N) if N is a scalar

DESCRIPTION
Generates binomial coefficients.

Example:
k=bincoef(5,15)

bino_rnd_____binomial random draws

CALLING SEQUENCE

[rnd]=bino_rnd(N,p,r,c)

PARAMETERS

INPUT:
* N = the # of trials (scalar)
* p = the probability of success (scalar)
* n = number of trials (scalar)
* r,c = size of random sample from binominal(N,p) distribution (optional; default=1,1)

OUTPUT:
rnd =(rxc) matrix of random draws from the binomial (N,p) distribution, equal to the number of successes in n-trials

DESCRIPTION
Random sampling from a binomial distribution.
NOTE: mean = a/(a+b), variance = ab/((a+b)*(a+b)*(a+b+1))

Examples:
1) x= bino_rnd(15,0.25,4,3)
2) x= bino_rnd(15,0.25)

Example 1 provides a (4x3) matrix of random draws from the binomial(5,15) law. Example 2 provides a scalar drawn for the same law.

chis_pdf_____chi-squared probability distribution

CALLING SEQUENCE

[f]=chis_pdf(x,a)

PARAMETERS

INPUT:
* x = a (nxm) matrix
* a = a scalar or a (nxm) matrix

OUTPUT:
a matrix of pdf at each element of x from chisq(a) or chisq(same element of a) distribution

DESCRIPTION
Returns the pdf at x of the chisquared(n) distribution.
NOTE: chis_pdf(x,n) = gamm_pdf(x/2,n/2)/2

Examples:
1) y=chis_pdf([1:4],2)
2) y=chis_pdf([1:4],2*ones(1,4))

Examples 1 and 2 do exactly the same thing: provide the (1x4) vectors of probabilities that a chi-square is greater than 1, 2, 3 and 4 respectively.

chis_rnd_____chi-squared random draws

CALLING SEQUENCE

[rn]=chis_rnd(nn,v)

PARAMETERS

INPUT:
* nn = a scalar for the size of the vector to be generated or nn(1) = nrows, nn(2) = ncols for a matrix to be generated
* v = the degrees of freedom

OUTPUT:
rn = (nxm) vector of draws from the chi-squared law

DESCRIPTION
Generates random chi-squared deviates.

Examples:
1) x=chis_rnd(18,2)
2) x=chis_rnd([4 2],3)
Example 1 creates a (18x1) vector of draws from the chi-square with 2 degrees of freedom distribution. Example 2 creates a (4x2) matrix of draws from the chi-square with 3 degrees of freedom distribution.

fdis_pdf_____Fisher probability distribution

CALLING SEQUENCE

[pdf]=fdis_pdf(x,a,b)

PARAMETERS

INPUT:
* x = a vector
* a = numerator dof
* b = denominator dof

OUTPUT:
* a vector of pdf at each element of x of the F(a,b) distribution

DESCRIPTION
Returns probability distribution function at x of the F(a,b) distribution

Example:
x=fdis_pdf([2.1 3.4 5.1],2,52)

Returns the vector of probabilities that a Fisher(2,52) is greater than 2.1, 3.4 and 5.1 respectively.

fdis_rnd_____Fisher random draws

CALLING SEQUENCE

[x]=fdis_rnd(n,a,b)

DESCRIPTION
return random draws from the F(a,b) distribution

PARAMETERS

INPUT:
* n = size of vector
* a = scalar dof parameter
* b = scalar dof parameter

OUTPUT:
x = a vector of random draws from the F(a,b) distribution

DESCRIPTION
Returns random draws from the F(a,b) distribution
NOTE: mean should equal (b/a)*((a/2)/(b/2-1))

Example:
x=fdis_rnd(100,2,18)

gamm_pdf_____Gamma probability distribution

CALLING SEQUENCE

[f]=gamm_pdf(x,a)

PARAMETERS

INPUT:
* x = variable matrix (nxm)
* a = a scalar or a (mxn) variable matrix

OUTPUT:
f = a vector of pdf at each element of x of the gamma(a) or of the gamma(corresponding element in a) distribution

DESCRIPTION
Returns the probability distribution function f at x of the gamma(a) distribution.

Examples:
1) f=gamm_pdf([1:4],2)
2) f=gamm_pdf([1:4],2*ones(1,4))

Examples 1 and 2 do the same thing: calculate the gamma(2) probability distribution function at values 1, 2, 3 and 4 and store them in a (1x4) vector.

gamm_rnd_____Gamma random draws

CALLING SEQUENCE

[gb]=gamm_rnd(nrow,ncol,m,k)

PARAMETERS

INPUT:
* nrow,ncol = the size of the matrix drawn
* m = a parameter such that the mean of the gamma = m/k
* k = a parameter such that the variance of the gamma = m/(k^2)

OUTPUT:
* r = an nrow x ncol matrix of random numbers from the gamma distribution

DESCRIPTION
Returns a matrix of random draws from the gamma distribution.
REFERENCE: Luc Devroye, Non-Uniform Random Variate Generation, New York: Springer Verlag, 1986, ch 9.3-6.
Note: m=r/2, k=2 equals chisq r random deviate

Example:
x=gamm_rnd(5,4,3,3)

hypg_cdf_____hypergeometric cumulative distribution

CALLING SEQUENCE

[p]=hypg_cdf(k,n,K,N)

PARAMETERS

INPUT:
k,n,K,N are the parameters as described above or matrix of parameters with the same size (pxq)

OUTPUT:
p = a (pxq) matrix of cumulative probabilities from the distribution

DESCRIPTION
Hypergeometric cdf function where the hypergeometric distribution corresponds to the probability of having k elements belonging to a class of size K in a random draw of n elements chosen among N.

Example:
p=hypg_pdf(5,15,20,30)

hypg_inv_____hypergeometric quantile

CALLING SEQUENCE

[k]=hypg_inv(p,n,K,N)

PARAMETERS

INPUT:
* p = a vector
* n,K,N = parameters of the hypergeometric distribution (n = size of the population drawn, K = size of the class of interest among the whole population, N = size of the whole population).

OUTPUT:
The vectors of the smallest integers k so that P(X <= k) >= p(i) for each coordinate i of p.

DESCRIPTION
Hypergeometric inverse (quantile) function.

Example:
x = hypg_inv([1:19]/20,15,20,50)

hypg_pdf_____hypergeometric probability distribution

CALLING SEQUENCE

[p]=hypg_pdf(k,n,K,N)

PARAMETERS

INPUT:
k,n,K,N are the parameters as described above or matrix of parameters with the same size (pxq)

OUTPUT:
pdf = (pxq) vector containing pdf for each x

DESCRIPTION
Hypergeometric pdf function where the hypergeometric distribution corresponds to the probability of having k elements belonging to a class of size K in a random draw of n elements chosen among N.

Example:
p=hypg_pdf(5,15,20,30)

hypg_rnd_____hypergeometric random draws

CALLING SEQUENCE

[x]=hypg_rnd(nobs,n,K,N)

PARAMETERS

INPUT:
* nobs = number of draws
* n,K,N = parameters of the distribution

OUTPUT:
x = a vector of random draws from the distribution

DESCRIPTION
Hypergeometric random draws.

```
prob(X=x) =   (N)^-1 (K)      (N-K)
              (n)    (x)      (n-x)
```

NOTE: mean = (n/N)*k ; variance = [(n*K)/(N*N)*(N-1)]*(N-K)*(N-n)

Example:
p=hypg_pdf(100,15,20,30)

logn_cdf_____lognormal cumulative distribution

CALLING SEQUENCE

[pdf]=logn_cdf(x,a,v)

PARAMETERS

INPUT:
* x = variable vector (nx1)
* a = mean vector (default=0)
* v = variance vector (default=1)

OUTPUT:
pdf == (nxm) vector containing cdf for each x

DESCRIPTION
Computes cdf of the lognormal distribution.

Example:
1) x=logn_cdf([0.2 0.5 1.5],2*ones(1,3),3*ones(1,3))
2) x=logn_cdf([1:5]')

Example 1 provide a (1x3) vector of the cumulative distribution function of a lognormal law with mean log(2) and variance 3 (for the log of the law) at values 0.2, 0.5 and 1.5. Example 2 provide a (5x1) vector of the cumulative distribution function of a lognormal law of mean 0 and variance 1 (for the log of the law) at values 1, 2, 3, 4 and 5.

logn_inv_____lognormal cumulative distribution

CALLING SEQUENCE

[linv]=logn_inv(x,a,v)

PARAMETERS

INPUT:
* x = variable vector (nx1)
* a = mean vector (default=1)
* v = variance vector (default=1)

OUTPUT:
linv == (nxm) vector containing pdf for each x

DESCRIPTION
Computes the inverse cdf (quantile) of the lognormal distribution.

Example:
1) x=logn_inv([0.1 0.5 0.9],2*ones(1,3),3*ones(1,3))
2) x=logn_inv([1:9]'/10)

Example 1 provide a (1x3) vector of the cumulative distribution function of a lognormal law with mean log(2) and variance 3 (for the log of the law) at values 0.1, 0.5 and 0.9. Example 2 provide a (9x1) vector of the cumulative distribution function of a lognormal law of mean 0 and variance 1 (for the log of the law) at values 0.1, 0.2,…0.9.

logn_pdf_____lognormal probability distribution

CALLING SEQUENCE

[pdf]=logn_pdf(x,a,v)

PARAMETERS

INPUT:
* x = variable matrix (nxm)
* a = mean matrix (nxm) or 1 (default=1)
* v = variance matrix (nxm) (default=1)

OUTPUT:
pdf == (nxm) vector containing pdf for each x

DESCRIPTION
Probability distribution function of the lognormal distribution for each component of x with mean a, variance v.
NOTES:
* the logarithm of a lognormal random deviate is normally distributed with mean = a and variance = v
* could probably be improved and simplified in the line of what is made in gamma_pdf

Examples:
1) x=logn_pdf([0.2 0.5 1.5],2*ones(1,3),3*ones(1,3))
2) x=logn_pdf([1:5]')

Example 1 provide a (1x3) vector of the probability distribution function of a lognormal law with mean log(2) and variance 3 (for the log of the law) at values 0.2, 0.5 and 1.5. Example 2 provide a (5x1) vector of the probability distribution function of a lognormal law of mean 0 and variance 1 (for the log of the law) at values 1, 2, 3, 4 and 5.

logn_rnd_____lognormal random draws

CALLING SEQUENCE

[r]=logn_rnd(mu,sigma,m,n)

PARAMETERS

INPUT:
* mu = the mean (may be a matrix)
* sig = the standard deviation (may be a matrix)
* m,n = the size of r in the case of mu or sig a matrix

OUTPUT:
rnd = a matrix of random numbers from the lognormal distribution

DESCRIPTION
Random draws from the lognormal distribution.
REFERENCES:
Evans, Merran, Hastings, Nicholas and Peacock, Brian, """"Statistical Distributions, Second Edition"""", Wiley, 1993 p. 102-105.

Example:
x = log_rnd(2,10,1000,1)

logt_cdf_____logistic cumulative distribution

CALLING SEQUENCE

[cdf]=logt_cdf(x)

PARAMETERS

INPUT: x = a (rxc) matrix or scalar argument

OUTPUT:
cdf = the (rxc) matrix of cdf of the logistic distribution

DESCRIPTION
Cumulative distribution function of the logistic distribution.

Example:
x=logt_cdf([1.5 ; 2 ; 2.5]

logt_inv_____logistic quantiles

CALLING SEQUENCE

[invl]=logt_inv(x)

PARAMETERS

INPUT:
x = a (rxc) matrix or scalar argument

OUTPUT:
invl = the (rxc) matrix of corresponding inverses (quantiles) of the logistic distribution

DESCRIPTION
Inverse of the logistic distribution.

Example:
x=logt_inv([0.1 0.2 ;0.3 0.4])

logt_pdf_____logistic probability distribution

CALLING SEQUENCE

[pdf]=logt_pdf(x)

PARAMETERS

INPUT: x = a (rxc) matrix or scalar argument

OUTPUT:
pdf = the (rxc) matrix of pdf of the logistic distribution

DESCRIPTION
Pdf of the logistic distribution at x.

Example:
x=logt_pdf([1 2 3 4])

logt_rnd_____logistic random draws

CALLING SEQUENCE

[rnd]=logt_rnd(r,c)

DESCRIPTION
random draws from the logistic distribution

PARAMETERS

INPUT:
r,c = size of the matrix, vector or scalar of draws

OUTPUT:
rnd = a matrix of random numbers from the logistic distribution

Example:
x=logt_rnd(5,2)

norm_pdf_____normal distribution

CALLING SEQUENCE

[pdf]=norm_pdf(x,m,v)

PARAMETERS

INPUT:
* x = variable vector (rxc)
* m = mean vector (default=0)
* v = variance vector (default=1)

OUTPUT:
pdf == (rxc) vector containing pdf for each value of x

DESCRIPTION
Computes the normal probability density function for each component of x with mean m, variance v.

Example:
x=norm_pdf([1 2; 3 4])

norm_rnd_____normal random draws

CALLING SEQUENCE

[y]=norm_rnd(sig)

PARAMETERS

INPUT:
* sig = a square-symmetric covariance matrix
* n= # of vectors drawn

OUTPUT:
y = (nxm) matrix composed of n (1xm) random vector normal draws with mean 0, var-cov(sig)

DESCRIPTION
Random multivariate random vector based on (mxm) var-cov matrix sig.
NOTE: for mean b, var-cov sig use: b + norm_rnd(sig)

Example:
x=norm_rnd([1 0.5 ;0.5 2],1000)

pois_pdf_____Poisson distribution

CALLING SEQUENCE

[pdf]=pois_pdf(x,l)

PARAMETERS

INPUT:
* x = variable vector (nxm)
* l = value at which to evaluate pdf

OUTPUT:
pdf == (nxm) vector containing pdf for each x

DESCRIPTION
Computes the normal probability density function at x of the Poisson distribution with parameter lambda.

Example:
x=pois_pdf(5,3)

pois_rnd_____Poisson random draws

CALLING SEQUENCE

[y]=pois_rnd(n,lambda)

PARAMETERS

INPUT:
* lambda = mean used for the draws
* n = # of draws

OUTPUT:
y = a vector of draws

DESCRIPTION
generate random draws from the poisson distribution
NOTES: Uses naive inversion method.

mean = lambda, variance = lambda

Example:
x=pois_rnd(10,2)

stdn_pdf_____Student distribution

CALLING SEQUENCE

[pdf]=stdn_pdf(x)

PARAMETERS

INPUT:
x = variable matrix (nxm)

OUTPUT:
pdf == (nxm) vector containing pdf for each x

DESCRIPTION
Computes the standard normal probability density for each component of x with mean=0, variance=1.

Example:
x= stdn_pdf([1 2 3 6]')

tdis_pdf_____Student probability distribution

CALLING SEQUENCE

[pdf]=tdis_pdf(x,n)

PARAMETERS

INPUT:
* x = a matrix, vector or scalar
* n = a matrix or scalar parameter with dof

OUTPUT:
pdf = a matrix of pdf at each element of x of the t(n) distribution

DESCRIPTION
Returnq the probability distribution function at x of the t(n) distribution.

Example:
1) x=tdis_pdf(2+[-5:5]/10,56)
2) x=tdis_pdf(2*ones(5,1),[10 ; 100 ; 1000 ; 10000 ; 100000])

tdis_rnd_____Student random draws

CALLING SEQUENCE

[t]=tdis_rnd(n,df)

PARAMETERS

INPUT:
* n = size of vector
* df = a scalar dof parameter

OUTPUT:
t = a vector of random draws from the t(n) distribution

DESCRIPTION
Returns random draws from the t(n) distribution NOTE: mean=0, std=1.

Example:
x=tdis_rnd(100,25)

unif_rnd_____Uniform random draws

CALLING SEQUENCE

[rnd]=unif_rnd(n,a,b)

PARAMETERS

INPUT:
* a = scalar left limit
* b = scalar right limit
* n = number of draws (default = 1)

OUTPUT:
rnd = (nx1) draws from the normal distribution

DESCRIPTION
Returns a uniform random number between a and b.

Example:
x=unif_rnd(100,2,8)

wish_rnd_____ Wishard random draws

CALLING SEQUENCE

[w]=wish_rnd(sigma,v)

PARAMETERS

INPUT:
* sigma = symmetric pds input matrix
* v = degrees of freedom parameter

OUTPUT:
w = random wishart_n(sigma) distributed matrix

DESCRIPTION
Generates random wishart matrix.
REFERENCES: Gelman, Carlin, Stern, Rubin, Bayesian Data Analysis, (1995,96) pages 474, 480-481.

Example (see wish_d()):
x = rand(100,5,'n'); xpx = x'*x; w = zeros(5,5); for i = 1:1000; w = w+wish_rnd(xpx,10); end